TOWARDS ACHIEVING CMMI LEVEL 2 FOR SMALL AND MEDIUM SIZED SOFTWARE DEVELOPMENT ORGANIZATIONS

BY

ISMAIL MOHAMED HEMDAN KESHTA

A Dissertation Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

August 2016

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This dissertation, written by **ISMAIL MOHAMED HEMDAN KESHTA** under the direction of his dissertation advisor and approved by his dissertation committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND ENGINEERING.**

Dissertation Committee

Dr. Mahmood Niazi
(Advisor)

Prof. Shokri Selim
(Member)

Dr. Mohammad Alshayeb
(Member)

Dr. Marwan Abu-Amara
(Member)

Dr. Muhammad Elrabaa
(Member)

Dr. Adel F. Ahmed
Department Chairman

Prof. Salam A. Zummo
Dean of Graduate Studies

24/3/16
Date

DEDICATION

In the Name of Allah, the Most Gracious, the Most Merciful.

To

My parents, who opened the way for me to success

iii

ACKNOWLEDGEMENT

All praise be to Allah, Subhanahu-wa-Ta'ala, for his limitless blessing and guidance. May Allah bestow peace on his prophet, Muhammad (Peace and blessing of Allah be upon him) and his family.

All my appreciation and thanks to my thesis advisor, Dr. Mahmood Niazi, for his guidance and help all the way till the completion of this dissertation.

I would like also to thank my dissertation committee members, Dr. Shokri Selim,
Dr. Mohammad Alshayeb, Dr. Marwan Abu-Amara and Dr. Muhammad Elrabaa for their cooperation and constructive comments.

Last, but not least, thanks to all my colleagues and friends, who encouraged me a lot on my way to achievement of this work.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

Name:	**Ismail Mohamed Hemdan Keshta**
Title:	**Towards achieving CMMI Level 2 for small and medium sized software development organizations**
Major Field:	**Computer science and engineering**
Date of Degree:	**September 2016**

CONTEXT: There is a significant need to give careful consideration to requirements management (REQM) and the process and product quality assurance (PPQA) process areas of Capability Maturity Model Integration (CMMI), especially in the context of small and medium-sized software development organizations in order to assist such organizations in getting one step closer to achieving CMMI Level 2 certification.

OBJECTIVE: The objective of this research is to implement REQM and PPQA process areas, specifically for small and medium-sized software development organizations. In this study, a workflow model for each specific practice in REQM and PPQA has been developed. In addition, guidelines, templates, forms, and checklists have also been provided. Moreover, the evaluation of the models has been discussed.

METHOD: Data has been collected by exploring published research articles and high-level software process descriptions. Previous research works that dealt with the implementation of the CMMI Level 2 process areas have also been reviewed. In addition, research articles that provide guidance to software development organizations for implementing process areas of CMMI Level 2 in their environments have been considered. Furthermore, the evaluation of the proposed models was executed using an expert panel review process

RESULTS: After careful analysis of the collected data, we proposed the models for each specific practice in REQM and PPQA process areas. Each model was divided into core stages, and different activities associated with each stage were clearly indicated. The evaluation results showed that our proposed models satisfy ease of learning and ease of use, provide stakeholder satisfaction, and can be applied to small- and medium-sized software development organizations.

CONCLUSIONS: Based on the evaluation, we are confident that our proposed models are clear and easy to learn, follow, and use. Moreover, our models are applicable to small and medium-sized software development organizations. The proposed models can also assist small and medium-sized organizations in implementing these two process areas.

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND ENGINEERING
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
Dhahran – Saudi Arabia
September 2016

خلاصة الرسالة

الاســــــــم : **إسماعيل بن محمد بن حمدان قشطة**
عنوان الرسالــة : **نحو تحقيق مستوى 2 من نموذج نضج القدرات المدمج CMMI للمنظمات الصغيرة والمتوسطة الحجم الصانعة للبرمجيات**
التخصــــــــص : **هندسة وعلوم الحاسب الآلي**
تاريخ التـــخرج : **ذي الحجة 1437 هـ**

السيــــــــــــاق: هناك حاجة كبيرة إلى النظر بعناية في إدارة متطلبات (REQM) وعملية ضمان جودة المنتج (PPQA) من نموذج نضج القدرات المدمج (CMMI)، خاصة في سياق المنظمات الصغيرة والمتوسطة الصانعة للبرمجيات من أجل مساعدة هذه المنظمات في الحصول على خطوة واحدة نحو تحقيق شهادة مستوى 2 من نموذج نضج القدرة المدمج CMMI.

الهـــــــــــــدف: الهدف من هذا البحث هو تنفيذ المناطق العملية REQM و PPQA، تحديدا للمنظمات الصغيرة والمتوسطة الحجم الصانعة للبرمجيات. في هذه الدراسة، تم تطوير نموذج العمل لكل الممارسات المحددة في REQM وPPQA. إضافة إلى ذلك، لقد تم توفير مبادئ توجيهية و قوالب وقوائم المراجعة. كما تم ايضا مناقشة لتقييم النماذج المقترحة.

الطريقــــــــــــة: لقد تم جمع البيانات من خلال استكشاف المقالات والبحوث المنشورة والأوصاف عملية البرمجيات. كما تم استعراض أعمال البحوث السابقة التي تناولت تنفيذ المناطق العملية الموجودة في مستوى 2 من نموذج نضج القدرة المدمج CMMI. وبالإضافة إلى ذلك، تم النظر المقالات والبحوث التي تقدم توجيها لمنظمات تطوير البرمجيات لتنفيذ المناطق العملية الموجودة في مستوى 2 من نموذج نضج القدرة المدمج CMMI في بيئاتها. وعلاوة على ذلك، تم تنفيذ تقييم النماذج المقترحة باستخدام عملية مراجعة لجنة خبراء.

النتائــــــــــــــج: بعد تحليل دقيق للبيانات التي تم جمعها، اقترحنا نماذج لكل ممارسة محددة في المناطق العملية REQM وPPQA. تم تقسيم كل نموذج الى مراحل أساسية، كما تمت الإشار بوضوح الى الأنشطة المختلفة المرتبطة بكل مرحلة. وأظهرت نتائج التقييم أن النماذج المقترحة لدينا تلبي سهولة التعلم وسهولة الاستخدام، وتوفير رضا أصحاب المصلحة، ويمكن تطبيقها بواسطة المنظمات الصغيرة والمتوسطة الصانعة للبرمجيات.

الاستنتاجـــــــات: بناء على هذا التقييم، نحن واثقون من أن النماذج المقترحة لدينا هي واضحة وسهلة للتعلم، والمتابعة، والاستخدامها. وعلاوة على ذلك، النماذج المقترحة قابلة للتطبيق في المنظمات الصغيرة والمتوسطة

الصانعة للبرمجيات. كما يمكن للنماذج المقترحة أيضا أن تساعد المؤسسات البرمجية الصغيرة والمتوسطة الحجم في تنفيذ تلك المناطقين العمليتين في نموذج نضج القدرة المدمج CMMI .

درجة الدكتوراه في هندسة وعلوم الحاسب الالي
جامعة الملك فهد للبترول والمعادن
الظهران- اللملكة العربية السعودية
ذي الحجة 1437 هـ

CHAPTER ONE: INTRODUCTION

This chapter introduces the problem statement and motivation for the study. Furthermore, it highlights the research questions that we plan to investigate and summarizes the contribution of this research.

1.1 The problem statement

Due to the significant increase in the importance of software products as well as the demand for enhanced software quality in the industry, many software engineering researchers pay special attention to the process of software development. Continuous assessments and improvements are needed throughout the software process to meet and satisfy the requirements of both customers and stakeholders. These improvements will lead to the creation of high-quality software. The quality of the software processes used by organizations for the development of software has a strong influence on the quality of the software product (Halvorsen et al. [1]). Thus, acquiring high-quality software that meets the specified requirements of customers is one of the major challenges that software organizations face (Kitchenham et al. [2]; Scacchi [3]).

Many researchers focused on software quality by applying the concept of software process improvement (SPI) (Ashrafi[4]; García-Mireles et al. [5]). Ashrafi

[4]investigated the impact of SPI methodologies on software quality. García-Mireles et al. [5] indicated that when software development organizations implement SPI, they are seeking to enhance the quality of their resulting software. SPI is considered an essential aspect in optimizing the software development process, especially for small and medium-sized organizations (Iqbal et al. [6]; Rahmani et al. [7]). Niazi et al. [8] pointed out that the design of effective SPI implementation initiatives to help small and medium-sized organizations is considered as one of the main challenges in the software industry. Therefore, research efforts have been directed toward the implementation of SPI frameworks/standards to increase software quality and productivity (Dutra et al.[9]; Niazi [10]).ISO/IEC 15504 and Capability Maturity Model Integration (CMMI) are examples of well-known and established SPI standards. CMMI is the latest SPI model introduced by the Software Engineering Institute (SEI). According to Lee et al. [11], "CMMI has been widely researched."

As shown in Figure 1, the CMMI model consists of five maturity levels (MLs). The lowest level is Level 1, which represents a poorly controlled SE process, while the highest, Level 5, represents an advanced SE capability. Twenty-two process areas (PAs) are associated with the five levels, except Level 1. Each PA has a set of related practices that should be carried out to satisfy and achieve a set of well-defined goals. Reaching a certain level of maturity for a software development company is based on the satisfaction of all the goals of the PAs in that particular level and in all the lower levels (Day et al. [12]).

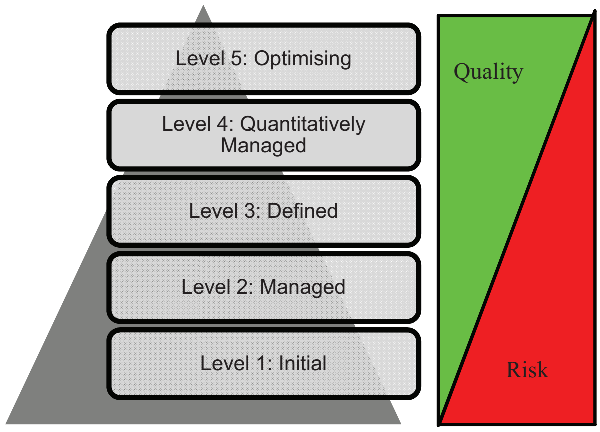

Figure 1: Capability Maturity Model Integration (CMMI)

Although the CMMI model assists software development organizations in enhancing their software quality, only a few software companies are adopting it. Staples et al. [13] highlighted significant reasons most software organizations do not use CMMI as an SPI model. The primary reasons are the small size of the organization, the long time needed to adopt such model, the high cost of services of capability maturity, and the use of other SPI models. Moreover, Niazi et al. [14] highlighted two main reasons many software organizations are unwilling to start a long path of CMMI implementation: the significant investment required and limited success.

It is important to note that small and medium-sized software development organizations are using ad hoc solutions in place of CMMI to implement SPI programs (Tarawneh et al. [15]). This is because such organizations experience more difficulty than larger organizations in applying Capability Maturity Models (Tarawneh et al. [15]; Clarke et al. [16]). In addition, they do not have enough significant resources to invest (Niazi et al. [14]). Chrissis et al. [17] stated that "CMMI identifies 'what' activities are expected, but does not specify techniques on how to accomplish those activities." Moreover, Vivatanavorasin et al. [18] pointed out that CMMI does not describe how the software development organization should act to achieve a given CMMI process area.

Thus, there are no clear approaches that assist small and medium-sized software companies in implementing CMMI. Therefore, more attention on "how" to implement CMMI is needed to assist such organizations in successfully adopting different CMMI levels. In this proposed research, CMMI Level 2 will be particularly addressed because Niazi et al. [8] pointed out that small and medium-sized software organizations set out to achieve Level 2 when they begin to accept CMMI as an SPI model.

1.2 Research objective

The objective of this research work is to implement requirements management (REQM) and process and product quality assurance (PPQA) process areas at CMMI Maturity Level 2, specifically for small and medium-sized software development organizations. The major contributions of this research are as follows:

- To develop a high-level (abstract-level) model for each specific practice in the REQM and PPQA process areas at CMMI Maturity Level 2, specifically for small and medium-sized software development organizations
- To evaluate the proposed workflow model for each specific practice in REQM and PPQA process areas through an expert panel review process

1.3 Research questions

In order to achieve these objectives, the following research questions (RQs) will be addressed:

- RQ1. How can one implement the specific practices in the REQM and PPQA process areas at CMMI Maturity Level 2?
- RQ2. What is the perceived "ease of learning and ease of use" of the proposed workflow model for each specific practice in the REQM and PPQA process areas?
- RQ3. What is the perceived "usefulness" of the proposed workflow model for each specific practice in the REQM and PPQA process areas?

- RQ4. What is the perceived “applicability” of the proposed workflow model for each specific practice in the REQM and PPQA process areas to small and medium-sized software development organizations?

1.4 Motivation

In this research, we will address Level 2 in CMMI because it has been noted that small and medium-sized software development organizations set out to achieve Level 2 when they start adopting the Capability Maturity Model as SPI standard to enhance the quality of their software product(Niazi et al. [8]).Particularly, REQM and PPQA process areas at CMMI Maturity Level 2 will be implemented, specifically for small and medium-sized software development organizations. The reasons for selecting these two process areas are as follows:

- **Requirements management (REQM) process area**

El Emam et al. [19] indicated that requirements engineering (RE) is considered one of the most critical processes and key tasks in software development. In other words, it plays a major role in the success of the software development process. Therefore, more attention should be given to improving requirements engineering practices. The StandishGroup [20]conducted a study in which they identified the reasons projects fail. They reported that a poor and inadequate RE process is one of the foremost reasons for project failure. The Standish Group [20] asserted that one of the major causes of project failure is the constant changing of requirements. The management of the changes in requirements in an effective and efficient manner can be employed as an essential predictor of project success. Research carried out by the Standish Group [72] indicated that requirements management and the change management process are important success points for the execution of small projects. In their study, they found that IT executives rated requirements management and change management on project performance as "very important" at 42% and 51%, respectively. Shah et al. [21] provided a review of significant issues and challenges of RE in the software development process, such as poor requirements traceability and requirements change management. The authors of this study pointed out that the important challenges of RE in software development remain unaddressed, even though industrial practices have proposed some solutions to overcome them.

- **Process and product quality assurance (PQA) process area**

Pacheco et al. [22] highlighted that quality, together with productivity, is a key factor in increasing competitiveness between different software development organizations. Quality plays a major role in averting a software project (such as a poor-quality software product). In addition, Pratt [23] indicated that poor quality implies potential costs, such as project failure, job loss, cost of reworking, and loss of opportunities. Furthermore, Young et al. [24] asked the question, "Why do so many projects fail?" in carrying out their study. The authors of this study were able to identify a list of the primary causes of project failure, including poor quality and inadequate quality control. Quality is an important factor that affects any software end product. Jarvis et al. [25] highlighted that PPQA is essential for both software development companies and those companies that buy the software. This is because PPQA gives indications if the techniques used during the project have been applied correctly. In other words, PPQA gives evidence that the procedures and techniques applied are integrated, properly implemented, and consistent.

In [8] and [26], Niazi et al. provided empirical studies to identify the relative perceived value of different specific practices of CMMI Level 2 process areas based on the experiences of practitioners in small and medium-sized companies. More than half of the practitioners cited the relative perceived value of all the specific practices of the PPQA process areas as medium. This means that the practitioners realized the importance of the PPQA process area. In other words, many participants involved in the study were aware of PPQA's importance. The findings also highlighted that less than 30% of the respondents regarded the value of all the specific practices of the PPQA process area as high. The authors provided two possible explanations for this finding. The first was that practitioners in small and medium-sized software development organizations paid limited attention and gave little consideration to PPQA activities. The second was that they had limited resources and a lack of technical expertise, which are both considered to be unique features of small and medium-sized software development organizations, compared with larger ones. Furthermore, the findings emphasized the major role that PPQA plays in achieving CMMI Level 2. Therefore, more attention should be given to PPQA to support the delivery of high-quality software products.

1.5 Summary of the research contribution

CMMI Level 2 was especially tackled in this research work because it was observed that small- and medium-sized software companies set out to achieve Level 2 when they begin to accept CMMI as an SPI model (Niazi et al.[8]). In particular, this work reported on the implementation of the REQM and PPQA process areas at CMMI Level 2, specifically for small and medium-sized software development organizations (reasons for selecting these process areas are highlighted in the previous section).

REQM is a basic project management process area at CMMI Maturity Level 2. The main aim of the REQM process area is to make sure the requirements are managed and consistent with each other throughout the conducted project. REQM process area involves five specific practices (see Table 1). In this work, an abstract-level model for each specific practice of the REQM process area was developed, with each model divided into core stages. Certain activities associated with each stage were described.

Table 1: Requirements Management (REQM) process area

CMMI Practice number	CMMI practice description
SP 1.1	Understand requirements
SP 1.2	Obtain commitment to requirements
SP 1.3	Manage requirements changes
SP 1.4	Maintain bidirectional traceability of requirements
SP 1.5	Ensure alignment between project work and requirements

PPQA is one of the fundamental process areas of CMMI Level 2. As stated in the CMMI v1.3 specifications, the purpose of PPQA is to "provide management and staff with objective insight into processes and related work products." the PPQA process area involves four specific practices (see Table 2), and achieving them greatly facilitates the delivery of high-quality software products. This is clearly because, throughout the software development life cycle, these practices offer objective insights to the software project managers. In addition, they provide appropriate feedback on both the software processes and related project work products. In this study, a workflow model for each specific practice in PPQA was developed. Each model was divided into core stages, and different activities associated with each stage were clearly indicated.

Table 2 Process and Product Quality Assurance (PPQA) process area

CMMI practice number	CMMI practice description
SP 1.1	Objectively evaluate processes
SP 1.2	Objectively evaluate work products
SP 2.1	Communicate and resolve noncompliance issues

SP 2.2	Establish records

In addition, a guidelines, templates, and checklists that can be utilized to traverse the proposed models for each specific practice of the REQM and PPQA process areas were provided. The evaluation of the proposed models was conducted using the expert review process. Moreover, the evaluation of these proposed models regarding “practice satisfaction,” “ease of learning,” “user satisfaction,” and “applicability to small and medium-sized software development organizations” was discussed.

The overall evaluation results showed that the proposed models ensured practice satisfaction according to the CMMI Maturity Level 2 requirement. They also satisfied the criteria for ease of learning and ease of use. Furthermore, they met the stakeholders' expectations and desired satisfaction level. Moreover, small- and medium-sized software development organizations can adopt the proposed models in their environments, as they were designed to be applicable to such organizations.

It is important to mention here that this research work will help small and medium-sized software development organizations to adopt CMMI Level 2 practices for the selected process areas in a quick manner. This will definitely help such organizations to be one step closer toward achieving CMMI Level 2 certification. In review, this work will improve the software performance and quality of a company that employs the use of CMMI Level 2 assessment in adopting CMMI Level 2 practices for the selected process areas. Also, it will increase the depth of knowledge about CMMI Level 2 for practitioners as well as researchers.

1.6 Thesis roadmap

The thesis is organized as follows: The background material of the research is presented in Chapter 2. Chapter 3 surveys related work in CMMI Level 2. Chapter 4 describes how our workflow models for each specific practice in the REQM and PPQA process areas at CMMI Level 2 were designed. Our findings from the literature for REQM are presented in Chapter 5. This chapter also describes the development of the models for the specific practices in REQM. In addition, it gives an evaluation of our models. Chapter 6 presents our findings from the literature for PPQA. This chapter also highlights the development and the evaluation of the proposed model for each specific practice in the PPQA process area. In Chapter 7, the limitations of the study are described. The conclusion and future work are also presented in this chapter.

CHAPTER TWO: BACKGROUND

2.1 Software Process Improvement (SPI)

Continuous assessment and improvement are required over the entire software development process to meet the expectations of stakeholders. Improving this process will result in the production of high-quality products. The quality of software products is strongly influenced by the quality of the software processes that the organization uses for development and maintenance. Therefore, one of the major challenges that software development organizations face is producing the desired quality of software products (Kitchenham et al. [1]; Scacchi [3]). According to Ashrafi [4] and García-Mireles et al. [5], much research has been carried out on software quality by focusing on SPI.

SPI is a mechanism that helps to tailor an organization's processes, and it is viewed as an essential aspect of optimizing the software development process. According to Nasir et al. [27], it is necessary to highlight that SPI implementation is strongly influenced, first, by organizational factors, such as political and cultural factors, and second, by project factors, such as budget, quality, tools, and technology. In addition, there are factors that play a key role in undermining the implementation of SPI programs. Niazi [28] identified and highlighted seven critical barriers (CBs) to the completion of such programs. These are as follows: inexperienced staff, lack of a defined SPI implementation methodology, lack of SPI awareness, lack of support, lack of resources, organizational politics, and time pressures. The author suggested that these CBs should be addressed by software development organizations when they develop SPI implementation initiatives. Unterkalmsteiner et al. [29] stated that "SPI is a systematic approach to increase the efficiency and effectiveness of a software development organization and enhance software products." Sommerville [30] mentioned that "process improvement means understanding existing processes and changing these processes to increase product quality and/or reduce costs and development time." Cugola et al. [31] pointed out that the ultimate goal of SPI is to increase both product quality and productivity. Not only that, but SPI has the ability to reduce time-to-market and production costs.

It can be noted that Unterkalmsteiner [29], Sommerville [30], Paulk [32], and Fox and Frakes [33], along with many other scientific researchers in this field, have emphasized that SPI is an important way for organizations to enhance the quality of their software products, increase their development productivity, accelerate their development processes, and reduce the cost and time needed to develop their software products. Niazi et al. [34] noted that designing appropriate SPI implementation initiatives is regarded as one of the main issues in the software industry, especially for small- and medium-sized software development organizations.

Hence, different SPI frameworks, standards, and models have been introduced to increase software quality and productivity (Iqbal et al. [6]; Rahmani et al. [7]). ISO 9000, PSP (Personal Software Process), CMM (Capability Maturity Model), and CMMI (Capability Maturity Model Integration) are all examples of well-known, established SPI standards that can be used by software development companies (Dutra et al. [9]; Niazi [10]).CMMI is the latest SPI model introduced by the Software Engineering Institute (SEI). According to Lee et al. [11], "CMMI has been widely researched." CMMI is a structured collection of the best practices used by organizations to assess the maturity of their software processes.

2.2 Capability maturity model integration-CMMI

CMMI is considered to be a very well-known Software Process Improvement (SPI) maturity model used to improve the quality of the software and its delivery. CMMI is a structured collection of the best practices used by organizations to assess the maturity of the software process.

Applying a CMMI assessment will help many types of software organizations to improve their processes and software product quality (Moser et al. [35]). Experts from three main areas —industry, government, and the SEI at Carnegie Mellon University—released this model in 2002. Based on data collected by the SEI, these experts believe that the maturity of software processes and the quality of the software products are highly interrelated Accordingly, both the software processes and software products need attention (O’Regan [27]; Monteiro et al. [36]).

CMMI can be considered as a framework that helps in assisting software organizations in the development of best practices. It is widely used throughout the world by large companies for process improvement, and thousands of organizations use such a model to enhance the quality of their software product. It is essential to highlight that the main aim of this model is to improve the software process. In other words, it can be used as a guide for improving the process across an organization. Thus, it ensures that the business needs are effectively met (O'Regan [27]). The CMMI model consists of five maturity levels (MLs). The highest level is level 5, which represents an advanced SE capability, while the lowest is Level 1, which represents a poorly controlled SE process.

2.3 CMMI Maturity Levels

The five CMMI maturity levels can be considered a roadmap for improvements in software organizations. In this section, a shot and brief overview of each level is highlighted.

2.3.1 Level 1: Initial

Level 1 is known as ad-hoc/chaotic level. This is basically because processes are disorganized, and there is no mechanism that can be used to enforce them. In this level, the basic characteristic of the software process is that it is informal and poorly controlled. Any organization that falls in this level is considered as an unstable environment for developing software. Thus, the processes at this level have unpredictable performance O'Regan [27]. The success of an organization that falls into Level 1 is most likely due to the heroics of people's individual efforts rather than followed processes. The weaknesses in the processes at this level will result in unpredictable cost and product quality (O'Regan [27]; Day et al. [12])

2.3.2 Level 2: Managed

The foremost characteristics of the software processes in this level are that Level 2 processes are planned, performed, and controlled. Any organization that falls into this category level has good project-management practices in place. The planning and management of the new project mainly depends on experience with similar, earlier-conducted projects that have similar applications (O'Regan [27]). Therefore, some of the processes followed in those successful earlier projects can be repeated again. In other words, some of the processes at this level are repeatable. Organizations that fall into this level have the basic project-management processes in order to monitor cost, schedule, and functionality (Hwang et al. [37]).

It is important to note that changes to work products and the project requirements in Level 2 are managed and controlled. This is clearly because of having appropriate configuration management control and project management policies [38].

Requirements Management, Project Planning, Project Monitoring and Control, Supplier Agreement Management, Measurement and Analysis, Process and Product Quality Assurance, and Configuration Management are key process areas (PAs) that will improve the performance of a company that is applying CMMI maturity Level 2 assessment. These PAs basically focus on two main categories, namely: the project management and support processes (Reitzig et al. [39]).

2.3.3 Level 3: Defined

Organizations that fall into CMMI maturity Level 3 have standardized and documented processes. Since they are well defined and described in standards, they can be easily understood. Moreover, they help in ensuring the consistency across the organization. Furthermore, all the specific and generic goals of PAs that are assigned to ML 2 and 3 are achieved by a company that is applying CMMI maturity Level 3 assessment. Moreover, wide training programs for the processes are provided by such organizations in order to build the required knowledge and skills for employees (O'Regan [27]).

In summary, at maturity Level 3, all projects must follow standards for developing and maintaining software. Requirements Development, Technical Solution, Product Integration, Verification, Validation, Organization Process Focus, Organization Process Definition, Organization Training, Integrated Project Management, Risk Management, and Decision Analysis and Resolution are key process areas that will improve the performance of an organization that falls into CMMI maturity Level 3 [38].

2.3.4 Level 4: Quantitatively Managed

Organizations that fall in this level are considered to be highly matured. At maturity level 4, processes are stable. In addition, they perform within pre-defined limits. Moreover, their performance is predictable. Both product quality and software process goals are understood, managed, and controlled by using statistical techniques. This level includes two PAs, namely: Organization Process Performance and Quantitative Project Management (O'Regan [27]).

2.3.5 Level 5: Optimized

Organizations at this level are also considered highly matured. They have continuous process improvement. This improvement is done based on a quantitative understanding of variation by using innovative ideas and technologies. There are two PAs at this level, namely: Organization Innovation and Deployment Causal Analysis and Resolution (O'Regan [27]).

In this model, there are twenty-two process areas which are associated with the five levels (except Level 1). Each process area has a group of related practices that should be carried out to achieve a set of feasible specific goals. The success in reaching a particular level of maturity for a software development organization is based on the fulfillment of all the goals of the process areas in that particular level and in all the lower levels (O'Regan [27]; Day et al. [12]).

There are two CMMI representations, namely: staged and continuous representations. Each representation uses the same PAs. Moreover, these two representations use the same specific and generic goals and practices. In addition, the level concept is the same for both of them. In other words, the same level concept is used regardless of which of the two representations is selected (Day et al. [12]).

2.4 Process areas of CMMI Maturity Level 2

There are seven key process areas in CMMI Level 2. These process areas are Requirements Management, Project Planning, Project Monitoring and Control, Supplier Agreement Management, Measurement and Analysis, Process and Product Quality Assurance and Configuration Management (O'Regan [27]). Such PAs will improve the performance of software development companies applying CMMI maturity Level 2 assessment. In this section, we will provide a short description on each of these PAs.

2.4.1 Requirements Management (REQM)

It can be seen that this PA is considered as basic project management (PM) responsibility. As in (Moser et al. [35]), the main aim of Requirements Management

(REQM) PA is to make sure that the requirements are managed and consistent with each other throughout the conducted project. In other words, they do not contradict each other. Moreover, REQM should keep the requirements up to date. In addition, such PA plays a major role in ensuring that requirements, project plans, and work products remain consistent (O'Regan [27]). REQM PA includes practices that mainly focus on the understanding of requirements, management, and control of changes to the software requirements (Moser et al. [35]). It is important to point out that this level includes REQM PA, but it does not introduce Requirements Development PA.

2.4.2 Project Planning (PP)

Establishing and maintaining project plans is the ultimate goal of Project Planning (PP). Such PA will fully set the foundation of the whole project. Throughout this PA, the software project is analyzed in terms of resources, work breakdown, and timing (O'Regan [27]). Crespo et al. [40] highlighted the difficulties that are inherent in the PP process area such as identifying projects, defining the life cycle, establishing a budget, and determining cost. There are three main specific goals for the PP process area, namely establishing estimates, developing a project plan, and obtaining commitment to the project plan. Achieving these goals means the software project is very well planned and established.

2.4.3 Project Monitoring and Control (PMC)

The main concern of project monitoring and control (PMC) is the monitor progress within the project. This means this process area highly depends on the deep understanding of the project's progress. Therefore, if the performance of the software project deviates from the plan, the PMC process area must take corrective actions (O'Regan [27]).

2.4.4 Supplier agreement management (SAM)

The selection of suppliers and documenting the statement of work with the supplier are the main focus of this PA. Moreover, SAM provides the management of acquisition of products from the supplier. Establishing and satisfying formal agreements with the suppliers are the specific goal for such a process area. Vivatanavorasin et al. [18] pointed out that failing in managing the supplier agreement management process area will result in encountering a time delay of the software product acquisition. Thus, the cost of the project will increase.

2.4.5 Measurement and Analysis (MA)

Determining the management information needs and measurement objectives are the ultimate objectives of the Measurement and Analysis (MA) process area. Thus, MA is mainly concerned with developing a measurement capability (McCabe [41]). The specific goals for this process area are, one, Align Measurement and Analysis

Activities, and two, Provide Measurement Results. Moreover, MA provides organizations with feedback on status of the conducted software project, the current state as well as the future state (O'Regan [27]; McCabe [41]).

2.4.6 Process and product quality assurance (PPQA)

As stated in [65], the purpose of PPQA is to "provide management and staff with objective insight into processes and related work products." Successful application of the PPQA process area will result in delivering high-quality software products. The specific goals for this process area are providing evaluating process service and providing objective insight service.

2.4.7 Configuration management (CM)

Establishing, maintaining, and managing the integrity of work products are the main purposes of Configuration Management (CM) PA. That means that CM PA is mainly concerned with the management of changes. This process area includes two specific goals, namely establishing baselines, tracking and controlling changes, and establishing integrity. Setting up configuration identification, configuration control, and configuration audits are involved in CM PA (O'Regan [27]).

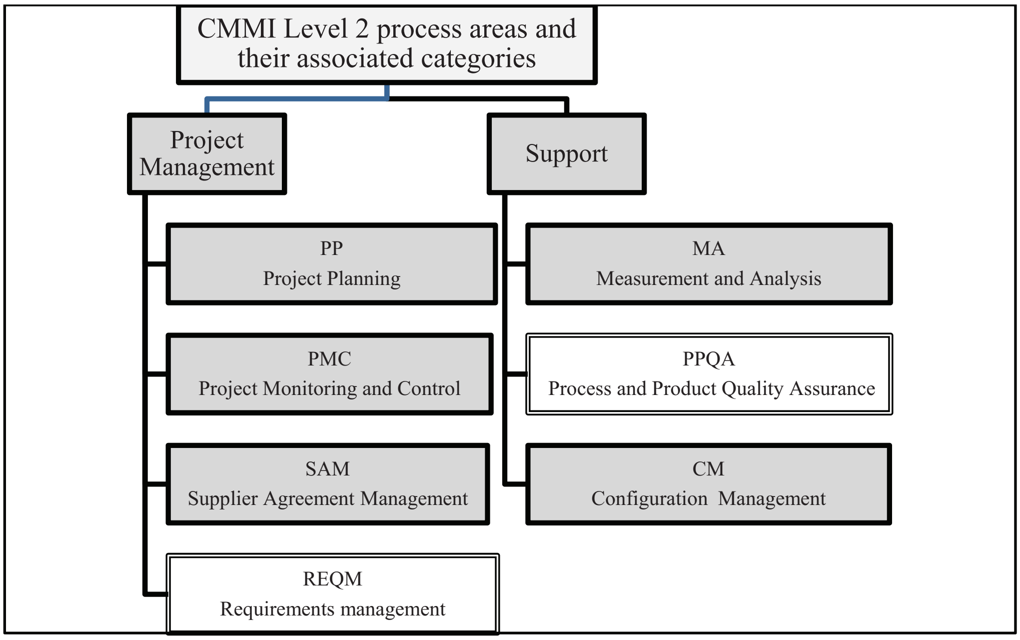

Figure 2: CMMI Level 2 Process Areas and their associated categories

As shown in Figure 2, the CMMI Process Areas (PAs) are classified into four categories, namely Process Management, Project Management, Engineering, and Support. Process areas in CMMI Level 2 can be classified into two categories, namely project management category and support category. Project Planning, Project Monitoring and Control, and Requirements Management and Supplier Agreement Management are under the project management category. On the other side, Measurement and Analysis, Process and Product Quality Assurance, and Configuration Management are under the support category.

2.5 Expert panel review process

Expert opinion approach (also known as expert judgment approach) is helpful in examining a series of specific questions related to the behaviour of the system, including its usability and reusability, as well as its performance (Babar et al. [120]). It is also utilized to carry out a product evaluation, which is done through a group of experts who are asked to use both their knowledge and experience in a certain area [121]. According to Nan et al. [122], the expert opinion's aim is to give a flexible, robust approach that will elicit unbiased evaluations from the domain experts. Garcia (2010) states that only one specialist is required during the elicitation process if it is proven to be perfect.

In addition, the valuable contribution of expert opinions has been widely recognised (Dyba [123]) in the SE community. Expert opinions in a specific matter can be explained as scientific efforts that are utilized to shed light on the data and predict the actions of a system, as well as assess the uncertainties (Cooke [131]). The increasing search for the views of experts particularly in pieces of academic research, has been justified because the decision-taking processes of many areas of knowledge are not mature yet or just being constructed (Li and Smidts [132]).

It is important to mention here that little samples can be utilized to trial, evolve a certain proposition, particularly in the early phases of a piece of research, according to Hakim [124]. Beecham et al. [125] confirms that the research uses samples to get feedback from experts in order to appraise the progress of models which will support an area of knowledge. Moreover, this approach is commonly used by many software engineering researchers to carry out an evaluation of various software development models. For instance:

- Dyba [123] utilised 11 SPI experts in both academia and industry to carry out the review process
- El Emam et al. [127] carried out interviews with a total of 30 experts to elicit the criteria in order to evaluate the success of RE.
- Beecham et al. [125] used 20 experts in order to validate their own Requirements Capability Maturity Model (R-CMM).
- Niazi et al. [87] introduced the requirements change management (RCM) model for implementing SP 1.3. This model was evaluated via an expert review process (i.e., two SPI expert reviewers involved in the evaluation)

Other work has also demonstrated that expert judgement is reliable, for example:

- Lauesen and Vinter [128] pointed out that the expert predications on the requirements defect were extremely high when they were used in practice. In other words, experts have a very good ability to predict the techniques that will ensure requirements defects are successfully prevented.

- Kitchenham et.al. [130] indicated that it is helpful to consider the opinion of experts on a process model in order to provide informal validation for the model.

It is important to highlight that the value of expert opinion is usually recognized when the quality of software is being assessed (Rosqvist et al. [129]). So, it can be stated that members of the software engineering community currently give a greater importance and more credibility to research that uses the technique of specialized opinion.

Other pieces of research have confirmed this assertion and also demonstrated how relevant this technique is, such as Kitchenham et al. [130], who have analysed the precision of a number of different methods of effort estimation by utilizing the various views of experts. Moreover, Beecham et al. [125] carried out research with experts that assessed a maturity model for software requirement engineering, which was aligned to the CMMI model.

As a result, researchers working in the software engineering sector tend to use specialist opinion in order to assess their proposal and get important feedback, which increases the efficiency of the evaluation process.

CHAPTER 3: RELATED WORK

3.1 Models for CMMI Level 2

3.1.1 Model for requirements change management

Niazi et al. [87] introduce the requirements change management (RCM) model for implementing specific practices in CMMI Level 2. This model is compatible with CMMI Level 2. It supports software development organizations in their RMC process. Their proposed model is based on two data collection sources. The first source is an extensive literature review of SPI and RE that includes research articles, published experience reports, and case studies. The aim of this source is to come up with a list of characteristics that effectively help in managing the requirements change process.

From the first source, three RCM models are identified for implementing specific practices in CMMI Level 2. These RCM models are Olsen's change management model, the spiral-like change management process, and Ince's change process model. The second source is interviewing experts in SPI from two companies. The second data collection source aims to identify the major interest of the two interview companies' representative in their RCM processes. The proposed model is based on five core stages: request, validate, implement, verify, and update. It is important to point out that the proposed model is validated by performing the evaluation using the expert review process.

3.1.2 Process model design for Supplier Agreement Management (SAM)

Vivatanavorasin et al. [18] stated that "CMMI defines Supplier Agreement Management process area (SAM) but it does not describe how the organization should do to achieve SAM process area." In this study, a business workflow process model for the SAM process area of CMMI L2 is presented. The proposed process model consists of three layers. These layers are known as the contextual layer, the elaboration layer, and the definition layer. In addition, designs need artifacts that include template documents; forms and checklists are also presented. Furthermore, Vivatanavorasin et al. developed a prototype tool called Supplier Agreement Management Tool (SAMT) to prove their proposed concept. The authors pointed out that their proposed model can guide software development organizations to implement the SAM process area of CMMI L2. It is necessary to note that Vivatanavorasin et al. clearly highlighted that software development organizations around the world have used CMMI practices as templates for improving their productivity and product quality.

3.2 Templates, process maps, and checklists for implementing CMMI Level 2 process areas

3.2.1 Process maps

O'Regan [27] presented a full chapter on the implementation of CMMI Level 2 in a software development organization. The main objective of CMMI Level2 implementation is to establish effective and successful management practices for software projects. Such implementation will enable practices that were used effectively and successfully in previously conducted projects to be replicated. Implementation of CMMI L2 means that the seven process areas (PP, PMC, REQM, SAM, PPQA, MA, and CM) need to be implemented. This may lead to seven separate process improvement teams that will work on the implementation of these PAs. The author highlights, however, that some of the improvement teams can be responsible for more than one PA. He provides the following example to illustrate this issue.

For instance, the improvement process team that will work on project management is traditionally responsible for PP and PMC. Moreover, this team may also be accountable for MA and SAM PAs as well. For REQM and CM process areas, separate process improvement teams are usually needed to implement these two PAs. For small software development organizations, O'Regan [27]pointed out that a software engineering process group (SEPG) may be the only improvement team that is accountable and responsible for implementing all process areas in CMMI Level 2. As an example, project planning, project monitoring and control, and measurement and analysis process areas of CMMI Level 2 will be presented in this literature.

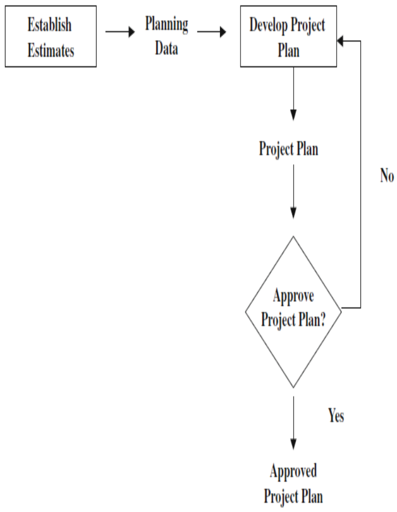

Figure 3 **: High-Level process map for PP**

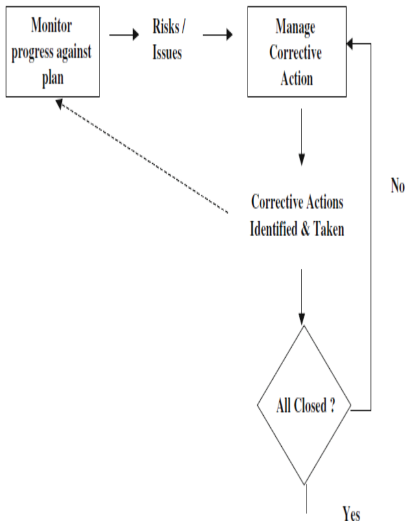

Figure 4 **: High-Level process map for PMC**

For project management, O'Regan [27] provides high-level process maps for project planning and project monitoring and control process areas of CMMI Level 2 because the project management improvement team is accountable and responsible for implementing these two PAs. The high-level process maps are given in Figure 3and Figure 4 . As shown in the figures, the high-level process map for PP establishes estimates, develops a project plan, and obtains a commitment to the plan. These are the specific goals of the project planning process area at CMMI Level 2. Establishing estimates includes the estimation of the scope of the project, work products and task attributes, project life cycle, effort, and cost.

O'Regan also provides one example of process maps for project planning process area. As shown in Figure 5, stakeholders and project managers will be involved in the process mapping for project planning and in the stream of activities. Project request; high-level estimates of budget, effort, and schedule; project authorization; preparation of the project plan; setting up the project board; defining project controls, identification of risks, preparation of the project schedule, project communication, project implementation; and project closure are highlighted in the provided sample process map for project planning.

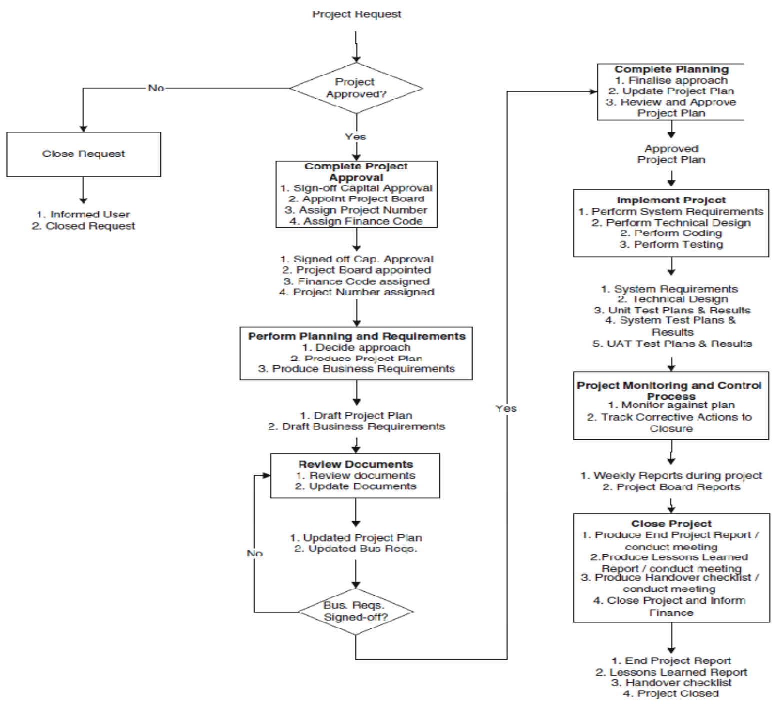

Figure 5: Sample process map for project planning

3.2.2 Templates and examples

O'Regan [27] states that "templates support the process and allow consistent input and output during the different parts of the process." As an example, the author provides typically needed templates (see Table 3) for implementing project management process that includes the implementation of project planning (PP) and project monitoring and controlling (PMC) process areas because the project management improvement team is accountable for implementing PP and PMC process areas in CMMI Level 2.

Table 3: Typical templates for project management

Template	Tool	Typical contents
Project plan	Microsoft Word	– Project goals and objectives – Estimation
Project schedule	Microsoft Project	Recording project tasks Recording effort and resources required

InterGlobe Consulting [95] is a famous organization that provides consulting and training services to bring project management to the global market. InterGlobe proposed a research paper entitled "A CMMI implementation case study." The ultimate aim of this work was to describe how software development companies can implement CMMI Level 2 project management processes to improve their project management methods. In this work, project planning (PP) and project monitoring and control (PMC) process areas in CMMI Level2 are addressed by transferring the specific goals of these two process areas into requirements for project management templates. In addition, examples of project planning and scheduling templates are provided. As an example, the paper illustrates the method for implementation. The specific practice (SP) 1.1 "estimates the scope of the project" in the project planning process area at CMMI Level 2. Three templates are needed to implement SP1.1. These templates are based on the product development lifecycle, the work breakdown structure (WBS), and the section of the PP that is relevant to the activity captured in the WBS.

In a master thesis entitled “Designing a Process Measurement Program as a Part of MA PA of CMMI Level 2” [94], the author proposed a measurement program that is based on the measurement and analysis process area at CMMI Level 2.This measurement program helps in collecting the numerical data and in data analysis and decision making. The design of the proposed measurement program is appropriate and applicable for repeatable process measurement.

The author noted that the design of a measurement process will help software development organizations toward process improvements. Furthermore, the author states that “the whole template would provide the organization with a guideline to achieve the organizational objectives.” In this work, a template is provided to show the status of the chosen project/process. It also helps in informing the stakeholders with necessary and basic information. The five phases of the process measurement are scope identification, definition of the process, data collection, data analysis, and involvement process. The main role of scope identification is to specify objectives supported by the measurement system and clearly identify obstacles that act against meeting those specified objectives. However, measures and metrics have to be specified. The second phase is defining the process. This phase includes the following tasks: clarifying operationally and consistently each identified measure, defining the data collection methods, and identifying the analytical techniques of the collected data. Data collection is the third phase in measuring the process. It involves three main tasks that are data collection, data recording, and data storage. The fourth phase is data analysis. The main role of this phase is to analyze the collected data and prepare the first draft of the reports. Furthermore, this phase also involves the continuous review of the whole procedure.

Finally, the last phase, that is the fifth phase, is process involvement. It enables the software development organizations to improve their processes. Framework for handling obstacles that act against meeting the objectives supported by the measurement system is provided by the final phase, process evolvement.

It is important to point out that the template is adjusted based on feedback from the academic side and stakeholders of Volvo 3P to produce an adjusted template “Volvo Version” for industrial use. In terms of validation, it is very important to highlight that the proposed template is validated and tested by Volvo 3P (industrial validation). Furthermore, one of the teachers at the Chalmers University of Technology worked on the validation and testing of this template to find out whether this template is appropriate in other fields other than the industry. The teachers found out that although this template was designed for Volvo, it was adjusted and accepted by Chalmers University of Technology. In other words, just minor changes were required in terms of wording and ordering of the questions in the template.

3.2.3 Checklist

O'Regan [27]states that "checklists are employed as an aid to performing the process." For example, the author provides a sample project management checklist that can help to ensure that the activities that are associated with PP and PMC process areas have been completed. The checklist is shown in the table below.

Table 4 Sample project management checklist

No.	Item to check
1	Is the project plan complete and approved?
2	Are the responses to the risks and issues appropriate?

3.3 Identifying high perceived value practices of CMMI Level 2

In [8]and [26], Niazi et al. address the challenges that are faced by small and medium-sized software development organizations in implementing CMMI Level 2 as SPI framework. They help software practitioners in such organizations to give more attention to the "high perceived value" of different practices of CMMI Level 2 process areas. This information enables such organizations to develop their own finer-grained CMMI Level 2 framework/standard, which will definitely help small andmedium-sized software development organizations to implement SPI initiatives in a better way.

Niazi et al. [26]conducted an empirical study in order to identify the high perceived value practices of CMMI Level 2. In their research, they only considered three PAs at CMMI Level 2: requirements management, process and product quality assurance, and configuration management. Establishing the perceived value of specific practices for these three PAs is the main contribution of this empirical study. Their work was based on face-to-face interview sessions with twenty-three Malaysian software development practitioners. Practitioners were asked to choose and rank the practices of three PAs against a five-point scale (high value, medium value, low value, zero value, or not sure). These five types of assessments help in identifying the level of the negative effect of a certain factor that works as a barrier against the successful implementation of SPI initiatives in Malaysia. Thus, they will help in indicating the importance of CMMI practices. Their results highlighted that Malaysian managers reported all specific practices of the REQM process area as high value. Malaysian developers, however, reported only one practice SP1.1 as high value. For PPQA, both Malaysian developers and Malaysian managers did not report any practice of this PA as high value. For the last PA, configuration management, Malaysian developers did not report any practice of this PA as high value. On the other hand, Malaysian managers reported four practices as high value. These four practices are SP1.2, SP1.3, SP3.1, and SP3.2.

Niazi et al.[8] did a substantially extended version of what was done in [26] by including practices of six CMMI Level 2 process areas instead of three. They also increased their sample size to include twenty-three from Malaysian software development practitioners and eight from Vietnamese practitioners. They followed exactly the same method as [26] in collecting data, which was to use a close-ended questionnaire based on CMMI Level 2 practices of six process areas. In this work, all PAs at CMMI Level 2 were included, except the Supplier Agreement Management process. This was because both Malaysian and Vietnamese software development participants were not managing the acquisition of products from suppliers. Their results show that in the REQM process area, Malaysian and Vietnamese practitioners reported three out of five specific practices as high value. The specific practices were SP1.1, SP1.2, and SP1.3. Malaysian and Vietnamese practitioners do not report project planning, project monitoring, control, measurement and analysis, process and product quality assurance, and configuration management as high value. It is important to point out that all specific practices of PPQA PA are reported medium value. Additionally, four out of seven specific practices are considered as having medium value. Similarly, the following eight out of fourteen specific practices of project planning are reported as medium value: SP1.1, SP1.3, SP2.2, SP2.3, SP2.4, SP2.5, SP2.6, and SP3.2.Out of ten specific practices of project monitoring and control practices, only four (i.e., SP1.1, SP1.2, SP1.4, and SP2.2) are reported as medium value. In this study, the authors highlighted the factors that work as barriers against the implementation of CMMI, which are resource constraints, company size, and time. Therefore, the implementation of CMMI is not easy for small and medium-sized software development companies, and this will definitely limit the benefits of CMMI as SPI standard even further.

3.4 Identifying measures for CMMI L2 process areas

3.4.1 Identifying measures for project planning (PP)

Khraiwesh in [115] addresses CMMI Level 2 by selecting one out of the seven process areas, the project planning process area. The ultimate objective of this study is to identify general measures for the three specific goals and its fourteen specific practices of the PP process area of CMMI L2. The author gives a set of questions and measures concerning specific practices from SP1 to SP14.

In terms of validation, a questionnaire is made to examine the validity of the defined measures. Software developers and designers in the information technology department of Zarqa University, Jordan, fill questionnaires using a scale of five options to collect their opinions. These options are as follows: strongly agree, agree, neither agree nor disagree, disagree, and strongly disagree. Table 5 gives an example of a questionnaire used to analyze the effect of defined sentences/measures on the accomplishment goal to estimate the scope of the project.

Table 5 Do you think that the following sentences/measures have an effect on the accomplishment of the goal of Estimate the Scope of the Project)

Statement serial	Statement	Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree

1	Establishing a top – level work breakdown structure (WBS) of the project.					
2	Defining the work packages in sufficient detail so that you can estimate project tasks, responsibilities, and schedule.					

3.4.2 Identifying measures for process and product quality assurance (PPQA)

Furthermore, in [96], Khraiwesh follows the same research methodology used in [115] to identify measures for the specific goals and their specific practices of the process and product quality assurance (PPQA) process area. That means that this study aims to address CMMI Level 2 by defining measures for two specific goals and its four specific practices of PPQA process area. A collection of questions and measures concerning specific practices (from SP1 to SP4) is provided in this research work. Similar to [115], the measures are determined by applying GQM to the specific practices of PPQA. A subset of questions and measures concerning practice 1 of PPQA at CMMI Level 2 are provided as an example.

Table 6: Subset of questions and measures concerning practice 1of PPQA in CMMI L2.

Questions	Measures
Do you produce evaluation reports? Do you produce evaluation noncompliance reports?	• Producing evaluation reports, noncompliance reports. • Number of Evaluation reports. • Number of Noncompliance reports.

The same form of validation is used in this research. For example, a questionnaire used to analyze the effect of defined sentences/measures on the accomplishment goal of "objectively evaluated processes" is provided in Table 7

Table 7: Do you think that the following sentences/measures have an effect on the accomplishment of goal of Objectively Evaluate Processes in PPQA process area in CMMI L2

statement serial	statement	Strongly Agree	Agree	Neither agree nor disagree	disagree	Strongly disagree
1	Evaluating selected performed processes.					
2	Producing evaluation reports, noncompliance reports.					

3.5 Questionnaire-based method for CMMI Level 2 process areas

Yucalar et al. [117] develop a questionnaire-based assessment method. The main objective of this method is to facilitate a quick assessment of CMMI Maturity Level 2 of a software development organization. In this study, the authors considered the software organizations that are new to CMMI. Moreover, software organizations that are just beginning to adopt CMMI as SPI standard to enhance their product quality are also considered. In other words, this study mainly targets small software development organizations. Yucalar et al. stated that "the method is not concerned at all with higher levels." The questionnaire-based assessment method consists of thirty-nine questions formulated to cover all seven process areas in CMMI Level 2:

REQM (3 questions), PP (9 questions), PMC (5 questions), SAM (4 questions), MA (5 questions), PPQA (5 questions), and CM (8 questions). It is indicated that the number of questions per process area is based on covering the range of practices per process area with as few questions as possible while maintaining the fairness among them. That is, the number of questions per process area does not reflect the importance of the process area. To each question, one out of five possible answers can be given: definitely yes = 4 points, usually = 3 points, planned but not applied = 2 points, not sure = 1 point, and definitely no= 0 point. The questionnaire-based assessment method is applied to five software companies in Turkey. Their result confirms the fact that the size of a software development organization plays a major role in its ability to achieve higher levels in CMMI.

3.6 Identifying factors affect transition time between CMMI levels

Alshammari et al. [113] perform an empirical study to indicate and investigate the factors that have a positive or negative effect on the transition time between CMMI levels in Saudi Arabia. The authors used a similar approach (Niazi and Babar, 2009). In this study, twenty-one factors were declared as effective factors and play a positive role for transition time between CMMI levels. The authors of this paper strongly recommend that Saudi software development organizations pay more attention to these effective factors in order to accelerate the transition time between CMMI levels. Thus, decreasing such transition will definitely lead software development organizations to business benefits. In addition, their results show that Saudi software development organizations need to improve their training, which can be achieved by offering training courses such as introduction to CMMI to stuff in their organizations. In addition, their study shows that out of twenty-one effective factors, “SPI implementation methodology” and “visibility into the SPI process planning” are two important effective and operative factors.

3.7 Description of the implementation of CMMI L2 best practices

Persse [64]addresses CMMI at Level 2 in his book entitled “Project Management Success with CMMI: Seven CMMI Process Areas.” He focused on how CMMI Level 2 can be applied to virtually any project. This book is strongly recommended for small and medium-sized software development organizations, because it helps them in implementing all seven CMMI Level 2 process areas. Additionally, the author introduced the intention of CMMI Level 2 process areas.

Author(s)	Year of Publication	Process Areas	Summary of Research
Models for CMMI Level 2			
Niazi et al.	2008	REQM	• Requirements Change Management (RCM) model is presented by implementing CMMI Level2 specific practice (SP1.3-1)“manage requirements changes practice.”
Vivatanavorasin et al. [18]	2006	SAM	• Process model for Supplier Agreement Management (SAM) process area of CMMI Level2 is presented. It helps software organizations in implementing SAM of CMMI Level2.
Identifying high perceived value practices of CMMI Level 2 process areas			
Niazi et al.[26]	2008	REQM, PPQA, CM	• The challenges that are faced by small and medium-sized software organizations in implementing CMMI L2 as SPI framework are highlighted. • The “high perceived value” of different practices of CMMI Level 2 process areas is pointed out to enable small and medium-sized organizations to develop their own finer-grained CMMI Level2 framework.
Niazi et al. [8]	2009	REQM, PPQA, CM, PP, PMC, MA	
Identifying measures for CMMI Level 2 process areas			
Khraiwesh [115]	2013	PP	• Measures for the three specific goals (SGs) and their fourteen SPs of PP of CMMI Level 2 are identified.
Khraiwesh [116]	2014	PPQA	• Measures for the two SGs and their four SPs of PPQA at CMMI Level 2 are identified.
Questionnaire based method for CMMI Level 2 process areas			
Yucalar et al. [117]	[117]2009	REQM, PPQA, CM, PP, PMC,MA, SAM	• A questionnaire based assessment method is developed in order to facilitate quick assessment of CMMI Level 2. The method is applied to five software companies in Turkey.
Templates, high-level process maps, and checklists for implementing CMMI Level 2 process areas			

InterGlobe[95]	2008	PP, PMC	• Implementation of PP and PMC of CMMI Level2 addressed by transferring the "SGs" of these two PAs into requirements for project management templates. • The example that illustrates the implementation of SP1.1"estimate the scope of the project" of PP PA at CMMI Level2 is presented.
Ghalambor [94]	2009	MA	• Important theories and methodologies such as literature study, meeting, and interview are used in designing the measurement template. The designed template is validated in academia as well as in the software industry.
O'Regan [27]	2011	REQM, PPQA, CM, PP, PMC, MA, SAM	• For each process areas, names of the templates and typical contents of these templates that are needed to implement PAs are given. Also, process map and checklist for each process areas are provided.
Identifying the factors affect transition time between CMMI levels in Saudi Arabia			
Alshammari et al.[113]	[113] 2011	———	• Factors that have a positive or negative effect on the transition time between CMMI levels in Saudi Arabia are identified. Discussion of the possibility of achieving CMMI L2 in six months is also provided. • Out of 21 effective factors, "SPI implementation methodology" and "visibility into the SPI process planning" are two important effective factors.
Describing the implementation of CMMI Level 2 best practices			
Persse [64]	2007	REQM, PPQA, CM, PP, PMC, MA, SAM	• The focus of how CMMI Level 2 can be applied to virtually any project is the main contribution. For each process area of CMMI Level 2, the description of how best practices can be implemented is provided.
Identifying specific practice dependencies in CMMI Level 2 process areas			
Xi et al.[114]	2008	REQM, PPQA, CM, PP, PMC, MA, SAM	• Introducing research work that identifies SP dependencies in process areas in CMMI L2 without changing the CMMI specification

3.8 Limitations in the existing studies

Although CMMI helps software development companies to improve the quality of their software, only a few of these organizations have adopted it. This is due to the small size of the organizations, the lengthy amount of time needed to adopt CMMI, and the high cost of CMMI's services (Staples et al. [13]). Many software development organizations often consider CMMI to be an expensive SPI model because it requires a significant amount of time to fully implement CMMI in their environments. Furthermore, Niazi et al. [14] indicated two major reasons that many software development organizations are unwilling to implement CMMI: the significant investment that is required and its limited success. Moreover, Batista et al. [42] indicated that small and medium-sized software development organizations, unlike larger ones, experience more difficulty with CMMI.

It is important to mention here that small and medium-sized software development organizations believe they are a long way from implementing SPI models such as CMMI because the implementation of such a model is difficult and expensive (Mejhem et al. [15]). Unlike larger companies, small and medium-sized companies do not do not have enough resources to invest. There is, therefore, a significant need to design CMMI Level 2 process areas to support and enable small and medium-sized software companies to quickly adopt CMMI Level 2 process areas.

As highlighted in the literature review, Eileen et al. [43] stated that "CMMI best practices describe what to do, but not how to do it." Vivatanavorasin et al. [18] also pointed out that CMMI does not describe how a software development organization should achieve a given CMMI process area. Furthermore, Chrissis et al. [17] stated that "CMMI identifies what activities are expected, but does not specify techniques on how to accomplish those activities." Gang et al. [44] also indicated that CMMI provides software organizations with general guidelines, but it does not provide them with a detailed operational model, so as a result, the cost of adopting CMMI is increased.

Since there are no clear and defined approaches that can provide help and support to small and medium-sized software companies in implementing CMMI Level 2 in an effective manner, further research into its effective implementation is required. In other words, there is a need for more concentration and careful consideration of how to implement CMMI Level 2, which is essential in assisting such software companies to increase their software quality and productivity.

Through our literature review, we have also noted that more attention should be paid to REQM and PPQA process areas. In other words, there is a need to give careful consideration to these two process areas and concentrate on them more.

Although REQM and PPQA process areas play important roles in the success of a software project, insufficient research studies have been performed on building effective implementation of these two process areas. Therefore, in this research, we have presented a workflow model for each specific practice in these two process areas especially in the context of small and medium-sized software development organizations in order to assist such organizations in getting one step closer to achieving CMMI Level 2 certification. Despite the importance of REQM and PPQA process areas, little research has been carried out on these two areas at the specific practice level. Research in this area is expected to provide useful insight into the implementation of these two process areas by developing an abstract-level model for each specific practice in these two areas.

Such implementation will help small and medium-sized software organizations adopt REQM and PPQA process areas quickly. Thus, it will assist them in getting one step closer to achieving CMMI Level 2 certification. In addition, this will lead to high-quality products and customer satisfaction. Furthermore, this implementation will increase the depth of knowledge of both practitioners and researchers on REQM and PPQA process areas at CMMI Maturity Level 2. Moreover, we can consider the research in this area as ongoing that can be extended and carried out by researchers in the future. It can possibly lead to research expansion in the field of CMMI Level 2 implementation

CHAPTER FOUR: RESEARCH METHODOLOGY

4.1 Stages involved in designing the proposed models

Figure 6 indicates the stages that are involved in designing the proposed models. The first stage involved in the development of the proposed workflow models is for specifying criteria for their success. Because of the unique features and peculiarities of small and medium-sized software development organizations compared with larger organizations, we have set the following basic criteria for developing the models for each specific practice in the REQM and PPQA process areas:

1. Ease of use: Small and medium-sized organizations do not have the money to adopt complex models. In addition, such organizations do not have the resources needed to apply complicated models. Therefore, our models have been structured to be easy to understand and easy to follow.
2. Stakeholders' satisfaction: The results of the models should satisfy the needs of the stakeholders. The proposed models should be utilized by the stakeholders to assist them in successfully attaining their specific goals and objectives.

3. Practice satisfaction: The proposed models should satisfy the CMMI v1.3 specifications. Each proposed model should satisfy the goal of the specific practice according to CMMI v1.3 specifications. In other words, the proposed models should ensure that there is practice satisfaction based on the CMMI Level 2 maturity requirement.
4. Applicability of the models to small and medium-sized software development organizations: The proposed models should be applicable to small and medium-sized software development organizations. In other words, they can be applied to both small and medium-sized software development organizations.

These criteria have been set because they were noted in the references cited in the literature (Niazi et al. [80]; Niazi et al. [87]). Thus, it was kept in mind that "the models should be easy to use, learn, and follow, and their results should satisfy stakeholders' needs" while building them. This is because we are targeting small and medium-sized software development organizations that face many traditional problems, such as lack of financing and lack of necessary knowledge.

In the second stage, the research questions were developed in order to meet these specified criteria. In the third stage, data was collected by exploring published research articles and high-level software process descriptions. In addition, previous research works that addressed the implementation of CMMI Level 2 process areas were reviewed. Moreover, research articles that provide guidance to software development organizations to implement process areas of CMMI Level 2 in their environments were considered. Then we carried out a very careful analysis and evaluation of the collected data. The proposed models were designed in stage 5. In the final stage, an evaluation of the proposed models was conducted with an expert panel review process.

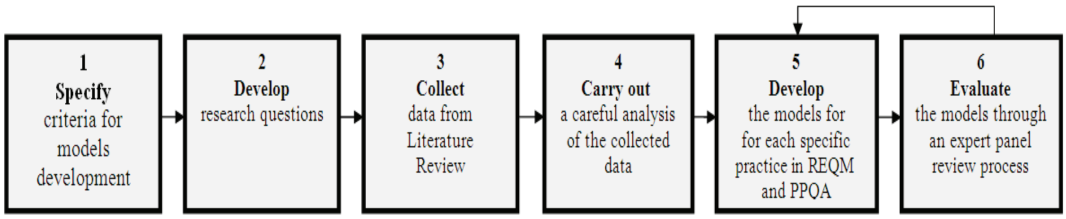

Figure 6 Stages involved in designing the proposed workflow models for each specific practice in the REQM and PPQA process areas at CMMI Maturity Level 2

4.2 Description of the stages

4.2.1 Stage 1

Setting the criteria for successful outcomes was the first stage involved in the design of the proposed models. The above basic criteria were used for building the models because of the unique and distinctive features of small and medium-sized software development organizations in comparison with larger ones. It is important to stress that the primary reason for carefully selecting the above criteria comes from the findings presented in Niazi et al. [118] and Niazi et al.[119]. Moreover, these criteria were specified as a result of the references reported in the literature from Niazi et al. [80] and Niazi et al. [87].

4.2.2 Stage 2

For the REQM process area, the following research questions (RQs) were addressed:

- RQ1. How can one implement the specific practices in the REQM process area at CMMI Maturity Level 2?
- RQ2. What is the perceived *ease of learning and ease of use* of the proposed model for each specific practice in the REQM process area?
- RQ3. What is the perceived *usefulness* of the proposed model for each specific practice in the REQM process area?
- RQ4. What is the perceived *applicability* of the proposed model for each specific practice in the REQM process area to small and medium-sized software development organizations?

For the PPQA process area, the following research questions (RQs) were addressed:

- RQ1. How can one implement the specific practices in the PPQA process area at CMMI Maturity Level 2?
- RQ2. What is the perceived *ease of learning and ease of use* of the proposed model for each specific practice in the PPQA process area?
- RQ3. What is the perceived *usefulness* of the proposed workflow model for each specific practice in the PPQA process area?

- RQ4. What is the perceived *applicability* of the proposed workflow model for each specific practice in the PPQA process area to small and medium-sized software development organizations?

4.2.3 Stage 3

For data collection sources, data was collected from the published literature by performing extensive literature reviews. The literature was conducted to help us in finding evidence from previous research work that addressed the implementation of CMMI Level 2 process areas. In addition, this step included exploring research articles, experience reports, high-level software process descriptions, and case studies.

4.2.4 Stage 4

The data collected from Stage 3 was carefully reviewed, analyzed, and evaluated. For each of the two selected CMMI Level 2 process areas, the aim of completing Stage 1 and Stage 2 was to have an identified list of characteristics that is required to effectively manage and control the selected process area. Moreover, models that can be adapted to implement the CMMI Level 2 practices for the selected process area were identified from the literature.

In this stage, each paper was reviewed carefully, and a list of guidelines and satisfactory models were identified from the literature on each specific practice. In papers in which the authors have described their CMMI implementation experiences with lessons learned, for example, how they achieved CMM Level 2, it was fairly simple to identify the guidelines and recommendations for the implementation of CMMI Level 2 process areas because often, authors provided a summary of these guidelines and recommendations as a bulleted list. However, in papers in which CMMI implementation was discussed but authors did not clearly provide a summary of findings about the main activities associated with REQM and PPQA process areas, each paper was carefully read to identify the core stages for implementing such process areas.

4.2.5 Stage 5

O'Regan [27] said, "Templates support the process and allow consistent input and output during the different parts of the process." Moreover, the author indicated that templates and checklists help ensure that the activities associated with process areas are completed. As indicated in the literature, Afrooz [94] said, "The template would provide the organization with a guideline to achieve the organizational objectives." Furthermore, [95] provides an example that presents how a specific practice in CMMI can be implemented by proposing templates, checklists, and forms. Therefore, an abstract-level model for each specific practice of the REQM and PPQA process areas was developed. Each model is divided into core stages, and different activities associated with each stage are clearly indicated. In addition, most important processes in the developed models for specific practices of the selected process areas were transferred into requirements for templates.

4.2.6 Stage 6

An expert panel review process was conducted to perform the evaluation of the proposed models. This process was employed to obtain the views of SPI experts regarding "practice satisfaction,""ease of learning and ease of use,""user satisfaction," and "applicability to small and medium-sized software development organizations" for the proposed models. Our selection of these experts was based on their experience in the field of software process improvement.

A questionnaire was created to obtain the experts' opinions regarding the models, and we extracted some questions from [93]. These were then tailored and customized so they were compatible with our research objectives. This questionnaire had three primary sections: a cover letter in which the objectives of the evaluation were described, demographics, and model feedback. Four important aspects were addressed in the last section of the questionnaire designated for the models' feedback. These aspects were "practice satisfaction," "ease of use," "user satisfaction," and "applicability to small and medium-sized software development organizations." We asked an academic researcher to examine the questions prior to sending out the questionnaire to the experts for evaluation, and it was determined from his response that some of the questions needed to be rewritten in order to produce a clearer, more concise, and more organized questionnaire that would better enable us to capture all the necessary data.

In short, the following points were considered in the evaluation forms:

- The objective of the questionnaire was described.
- Some questions were asked about the experts' experience and knowledge of CMMI implementation.
- The expert reviewers were asked to evaluate the proposed models against the "ease of learning and ease of use" criteria.
- The expert reviewers were asked to evaluate the proposed models against the "practice satisfaction" criteria.
- The expert reviewers were asked to evaluate the proposed models against the "user satisfaction" criteria.
- The expert reviewers were asked to evaluate the proposed models against the "applicability of the models to small and medium-sized software development organizations" criteria.
- The expert reviewers were asked if he/she would like to provide comments on how we could improve the proposed models. This point gave the expert reviewers the chance to respond freely and include more information.

4.3 Limitation of research design

This research study was limited in a number of ways, and it is important to express three of these limitations in particular.

1. The SPI expert reviewers were likely to have different views about each of the questions in the questionnaire and interpret them in completely different ways. In other words, the expert reviewers would have replied to questions according to their own interpretations. It is also quite possible that the expert reviewers misunderstood the description of the proposed models appended with the questionnaire and could have wrongfully interpreted the questions that were asked on each proposed model. It is important to note here that there was no noticeable evidence of this limitation, as none of the expert reviewers ever reported an issue or concern. However, in order to mitigate the effects of this limitation, the questionnaire was tested by an academic researcher before it was sent to the expert reviewers, and the final version was based on the feedback. This was done to ensure that each question was clear, and the expert reviewers would understand them all in the same way as intended.
2. The evaluation results were limited by the expert reviewers' knowledge and experiences, but we were confident about the results because all the

respondents that were involved in the evaluation process had the necessary knowledge and experiences in the fields of SPI and RE.

3. It is important to mention here that an ordinary literature survey was used as the main approach for data collection. However, if a systematic literature review was used for data collection, we might get better result because it provides in-depth and more thorough results than an ordinary literature review. However, we were able to reduce the impact of this limitation by performing academia evaluation.

CHAPTER FIVE: REQUIREMENTS MANAGEMENT (REQM)

5.1 Requirements Management (REQM) Overview

Requirements management (REQM) is a basic project management process area at CMMI Maturity Level 2. The main aim of the REQM process area is to make sure the requirements are managed and consistent with each other throughout the conducted project. In other words, they do not contradict each other (O'Regan [27]). Moreover, REQM process area should keep the requirements up to date. In addition, such PA plays a major role in ensuring that requirements, project plans, and work products remain consistent (Wilkie et al. [62]). REQM process area involves five specific practices.

Table 8: Requirements Management (REQM) process area

CMMI Practice number	CMMI practice description
SP 1.1	Understand requirements
SP 1.2	Obtain commitment to requirements
SP 1.3	Manage requirements changes
SP 1.4	Maintain bidirectional traceability of requirements
SP 1.5	Ensure alignment between project work and requirements

As described in Table 8, these practices mainly focus on the understanding of the requirements, obtaining commitment to the requirements, managing and controlling the changes to the requirements, maintaining bidirectional traceability of requirements, and identifying inconsistencies between project work and requirements.

- **SP 1.1 – "Understand requirements"**

CMMI Level 2 provides a key specific practice in REQM PA, entitled SP 1.1 "Understand Requirements." According to CMMI v1.3 specifications [65], this specific practice helps software development organizations successfully understand the requirements and the problem domain. It also ensures that all stakeholders have established a joint understanding of the requirements for the project, which will result in a successful software development project and satisfactory outcomes. Achieving and satisfying SP 1.1 will also result in an agreed set of requirements and a plan that satisfies both the customers and the users. In addition, such specific practice assists in completing the software development process within the given time and budget constraints (Persse [64]).

It is important to mention here that Niazi et al. [8] provided an empirical study to identify the relative "perceived value" of different specific practices of CMMI Level 2 PAs based on the experiences of practitioners of small- and medium-sized software development organizations in Malaysia and Vietnam. In this study, more than half of the practitioners reported the relative perceived value of SP 1.1 as "high." Lester et al. [63] also provided an empirical study in order to identify the best specific practices of CMMI Level 2 process areas followed by both small- and medium-sized software development organizations. Their findings showed that SP1.1 had a tendency to be the highest importance-specific practice to small organizations and the second highest to medium-sized organizations. In other words, SP 1.1 has the highest and second highest value in the average values for practice usage in small- and medium-sized software development organizations, respectively.

- **SP 1.2 – "Obtain commitment to requirements"**

One of the fundamental practices in REQM PA at CMMI Level 2 is SP 1.2. This practice mainly deals with obtaining commitment to the requirements. CMMI v1.3 specifications [65] highlight that SP 1.2 aims to control and monitor the agreements, commitments, and responsibilities between the involved persons/groups. Persse [64] provided three reasons why SP 1.2 is extremely important. First, it indicates that the work is intended to be continual and forward moving, based on the agreements between the involved persons/groups. It also helps in setting up and putting in place the agreements and concerns. Finally, it emphasizes that the involved persons/groups have a chance to not accept the proposed changes when they are asked to approve any changes that might happen after obtaining the commitment to the requirements.

It is important to mention here that Wilkie et al. [62] performed an empirical study of small- and medium-sized software development organizations in a bid to evaluate the CMMI process areas. The authors carried out research to explore the challenges such organizations had when adopting CMMI in their environments. They identified the perceived value for each specific practice of CMMI Level 2. Their findings showed that SP 1.2 had the highest importance-specific practice to such organizations. Niazi et al. [8] also indicated that SP 1.2 is often cited as "high" perceived value by Malaysian and Vietnamese software development participants in small- and medium-sized software development organizations. In addition, Lester et al. [63] pointed out that SP 1.2 had the highest average value for practice usage for REQM PA in medium-sized software development organizations and the second highest value in smaller organizations. Consequently, they suggested that more consideration and focus should be paid to SP 1.2.

- **SP 1.3 – "Manage requirements changes"**

SP 1.3 is one of the primary practices in REQM process area. This practice deals with managing the changes in the requirements. According to the CMMI v1.3 specifications [65], SP 1.3 is an essential practice in managing and controlling changes that may happen after obtaining commitment to the existing requirement. Persse [64] indicated that SP 1.3 is clearly defined and described to assist in managing the constant evolution of the requirements. Once any change is triggered through the requirements change management, the impact of the proposed change should be evaluated and analyzed based on the current commitments, work breakdown structure, project plan, and other work products. Any change in the requirements may immensely impact the software project. Thus, SP1.3 will help evaluate the change in requirements and analyze the impact caused with regards to the schedule, cost, and quality. In addition, this specific practice will assist in identifying the nature and complexity of changed requirements.

It is necessary to point out that Wilkie et al. [62] performed an empirical study of small- to medium-sized software development organizations in a bid to evaluate the CMMI process areas. The authors carried out research to explore the challenges such organizations had when adopting CMMI in their environments. Their findings showed that both small- and medium-sized organizations failed to show effective and efficient strategy plans in evaluating and analyzing the impact of proposed changes to the existing requirements. Niazi et al. [8] also provided an empirical study to identify the relative "perceived value" of different specific practices of CMMI Level 2 process area based on the experiences of practitioners of small- and medium-sized software development organizations in Malaysia and Vietnam. Their findings indicated that SP1.3 was often cited as "high." This result confirmed the prior findings of other research studies (Niazi [73]; Niazi et al. [74]) that emphasized the significant role of SP1.3 in managing the requirements of the software development process. Furthermore, Lester et al. [63] showed that, in terms of the average values for practice usage for REQM PA, SP1.3 had the least value in medium-sized software development organizations and was the second lowest in smaller ones.

- **SP 1.4 – "Maintain bidirectional traceability of requirements"**

One of the core specific practices in REQM process area at CMMI Level 2 is SP 1.4. According to the CMMI v1.3 specifications [65], SP 1.4 is an essential practice in maintaining and supporting the bi-directional traceability of requirements. Once the constant evolution of the requirements is controlled and managed, requirements traceability is required to be initiated in order to trace a source requirement to its resulting work product and to trace each unique work product requirement back to its source. CMMI v1.3 specifications also underline that SP 1.4 assists in deciding whether or not all source requirements are perfectly tackled. In addition, this specific practice helps in determining whether or not all low-level requirements can be mapped and verified to a valid source. In other words, such practice assists in finding the valid origin and root of all low-level requirements. It is necessary to indicate that CMMI v1.3 specifications extremely stress the importance of this practice, especially in the case of assessing the impacts of the requirements changes on the current commitments, work breakdown structure, project plan, and other work products. Persse [64] highlighted that SP 1.4 contributes to three main project management points: trace to plan, trace to anticipate, and trace to know.

It is important to mention here that Wilkie et al. [62] showed that most small- and medium-sized software development organizations fail to present effective and structured strategy plans in tracing the lifecycle of each requirement. Niazi et al. [8] also indicated that SP1.4 was reported as a "high" perceived value by only 41% of small- and medium-sized software development participants. In this study, the researchers stress that SP1.4 is one of the most essential elements of the requirements management process. This is because it is impossible to manage and control the requirements without implementing a well-defined traceability strategy, except for very small projects. Furthermore, Lester et al. [63] illustrated that SP1.4 was the least importance-specific practice to small-sized organizations and the second lowest to medium-sized organizations.

- **SP 1.5 – "Ensure alignment between project work and requirements"**

SP 1.5 is indicated to be one of the main specific practices in REQM PA at CMMI Level 2. According to the CMMI v1.3 specifications [65], this practice identifies the inconsistencies between the project plans and the work products and the requirements, and then resolves them by initiating corrective actions. In other words, SP1.5 assists in ensuring that the software project plans and work products are accurately reflecting the current state of the requirements. It is also considered a fundamental practice in keeping the requirements synchronized to software project plans and the work products produced across the various phases during project execution. Besides, this practice highlights the importance of proposing corrective actions that need to be performed to resolve the inconsistencies. Moreover, Persse [64] considered SP1.5 as the culmination of the other specific practices in requirement management process area at CMMI Level 2.

Persse [64] goes on further to highlighted two important benefits of applying SP 1.5, i.e., (1) harmony with plans and (2) harmony with work products. It is necessary to illustrate that the first benefit indicates that during the software development process, the software requirements may require modifications such as additions or deletions. This may result in significant changes in the software project plans. Therefore, the requirements, as well as the software project plans, should be kept aligned. While the second point highlights that keeping the project's work products aligned with the requirements is considered to be one of the significant activities of the requirement management. In other words, the progress of work products should be monitored and tracked to ensure that the developed work products of the software project reflect the present state of the requirements.

In addition to the above studies, there are various empirical studies (such as Wilkie et al. [62], Niazi et al. [8] and Lester et al. [63]) that were performed in order to address the challenges faced by small to medium software development organizations, in adopting CMMI Level 2. They also assist such organizations to pay more attention to the "high perceived value" of different CMMI Level 2 specific practices (including SP 1.5). Overall, the results of these studies emphasized the significant role of SP 1.5 in managing the requirements and also suggested to give more consideration and focus to SP 1.5.

It is important to mention here that Wilkie et al. [62] showed that 50% of the small- and medium-sized software development organizations failed to ensure that their plans and work products accurately reflected the present state of their requirements. This means that half of these organizations were unsuccessful in keeping the requirements updated and synchronized with the project plan and the work products that were developed during project execution. Thus, there is a need for the organizations to give more consideration to SP1.5. Another study presented by Niazi et al. [8] depicted that 48% of the respondents of small- and medium-size software development organizations in Malaysia and Vietnam reported SP1.5 as a 'high' value practice. In Lester et al. [63], the research findings indicated that SP1.5 was the third most important practice for both small- and medium-sized software development organizations.

It can be concluded that more attention should be paid to SP 1.1, SP 1.2, SP 1.3, SP 1.4 and SP 1.5 in REQM process area. In other words, there is a need to give careful consideration to these two specific practices and concentrate on them more. Although SP 1.1, SP 1.2, SP 1.3, SP 1.4 and SP 1.5 play important roles in the success of a software project, not enough research studies have been carried out to support the effective implementation of these two specific practices. Thus, in this research, we propose a model for SP 1.1, SP 1.2, SP 1.3, SP 1.4 and SP 1.5 in REQM process area.

5.2 Proposed Models

5.2.1 SP 1.1–“Understand requirements”

This section illustrates our findings, which include the guidelines and recommendations for understanding the requirements. It also provides the answer to RQ1 for SP 1.1. In addition, suitable models and process maps that would help develop an understanding of the requirements' providers with regard to the meaning of the requirements are presented. This section also describes the development of the proposed workflow model for SP 1.1"understand requirements."

5.2.1.1 Related work on understanding the requirements

Different research studies have been conducted to provide recommendations on better understanding the requirements:

- Azlena *et al.* [45] identified the five steps: (1) relation of software project life cycle with RE, (2) understanding the definition of requirements engineering, (3) proper selection of RE processes, (4) decision of RE deliverables, and (5) understanding RE practices.

- Saqib et al. [46] investigated the level of understanding of requirements engineering (RE) practices by performing an empirical study on software industry organizations in Pakistan. The authors identified the main issues and challenges that might be encountered by small and medium enterprises in Pakistan during the RE process. Their results highlighted the importance of domain understanding in achieving a better understanding of requirements.
- Persse [64] provided the following five activities to support and implement the understanding of requirements:
    1. Document the requirements: The main advantage of documenting the requirements is that the requirements document is shared and everyone can access the same set of information at any time. Microsoft Word is used by many RE teams to write, update, and modify their requirements document.
    2. Identify stakeholders: All people who might be impacted by a project should be identified. Also, those who have an impact on a project have to be clearly indicated. In addition, the external and internal stakeholders have to be identified. The interest and influence of each stakeholder can be determined by using stakeholder analysis. Stakeholder register is the main output of this activity.

3. Distribute the requirements for review: This activity can be performed by sending email with the attached requirements document to the stakeholders when the need arises. Also, the updated versions of the requirements document should be distributed among stakeholders from time to time.
4. Allow time for adequate review: To have a good understanding of the requirement, stakeholders need a reasonable amount of time to review the requirements document.
5. Encourage feedback: The comments of stakeholders should be given serious consideration. This will encourage stakeholders to understand the requirements and give more feedback. Online comment forms, email, and issue logs can be used in this activity.

- Moorthy [81] wrote a book titled, CMMI Implementation Guide: A Practitioner's Perspective. This book can be considered a reference book for software development organizations that have an interest in implementing CMMI in their environments. The author of this book proposed samples of typical artifacts that can be used in organizations, and they meet the CMMI practices. The typical CMMI artifacts proposed for SP 1.1are requirements document, collection of requirements, and mail.

- O'Regan [27] pointed out that achieving and satisfying SP 1.1 will result in an agreed set of requirements. In addition, criteria will be established and defined in order to assess and accept the requirements.
- Chrissis et al. [17] provided examples of evaluation and acceptance criteria, i.e. the requirements should be uniquely identified, clearly stated, complete, traceable, consistent with each other, and verifiable.

In addition to the above studies, some models and process maps to better understand the requirements have also been proposed:

- Rickman [47] proposed a model for understanding the requirement. This model is iterative and consists of 21steps.
- Spiral model of the RE process proposed by Kotonya and Sommerville [48] helps with developing a better understanding of the requirements through iteration. This model is used in the general RE process. It consists of requirements elicitation, requirements negotiation and analysis, requirements documentation, and requirements validation. In this model, the RE process is presented in an alternative way. In addition, the decision is made after the repetition of different RE activities. In other words, from one round to another, all stages of the process are repeated (Sommerville [49]). In each round, details may be added to the requirements. Thus, the requirements document is accepted and well understood. It is important to point out that

this model helps refine the understanding of the requirements. In this model, understanding the requirements can be considered a prerequisite for initial design work (Jeffrey [50]). Moreover, as Leffingwell [51] stated, “An initial pass around the spiral is intended primarily to understand requirements.”

- O'Regan [27] proposed a sample process map in which different activities help determine and validate customer requirements as well as product requirements. The proposed sample process map shows the flow of activities associated with REQM process area. In addition, analysis activities, such as conduction workshops and creation of business requirements, are also pointed out. Updating the requirements document activity, after the requirements validation, is also highlighted. Moreover, this process map shows the need for the following: business requirements to identify the customer’s needs and system requirements to clearly indicate what the software development organization is required to provide.

5.2.1.2 The proposed model for SP 1.1–“Understand requirements”

Our proposed model for SP1.1"understand requirements" is mainly based on our findings from the literature presented in Section 4.2.1.2. The model is divided into five essential stages: "Request," "Understand," "Evaluate," "Accept," and "Finalize." Certain particular activities associated with each of these stages are clearly indicated. Figure 7 shows the proposed workflow model for SP 1.1.

The first stage is the "Request" for new projects. This stage was included because the request process was found in the requirements process map proposed by O'Regan [27]. As highlighted in CMMI v1.3 specifications [65] for SP1.1, requirements should be provided by appropriate requirements providers based on certain criteria that are created to designate suitable channels for the receipt of requirements. Therefore, this stage in our model has been structured with a main process called "Provide Requirements." That means software requirements must be provided only by a designated user/customer to the software project team members/business analysts. The requirements can be provided in the form of a Statement of Work (SOW), as it is considered to be the most common form of providing requirements. In the requirement understanding model (Rickman et al. [47]), SOW was highlighted as a vital element in the first step to understanding the requirements. For SP1.1, Persse [64] highlighted that the main objective of understanding the requirements is to ensure that the people involved in the project are comfortable in order to come up with a set of approved and base-lined requirements. In order to effectively achieve this objective, distinguishing the stakeholders based on a list of certain criteria needs to be performed. Considering too many stakeholders will make the project difficult to understand. Therefore, the interest and influence of each stakeholder should also be determined.

Distinguishing stakeholders is a key step in gathering and understanding the requirements. It allows the project team to focus on understanding the areas that meet the needs of stakeholders. In addition, it helps prioritize the attention and management of multiple stakeholders' interests and their influence. In order to tackle the understanding of the requirements and the problem domain, stakeholders should be identified and distinguished, because it is difficult to understand the requirements for a project without first distinguishing who is affected. Therefore, if stakeholders are correctly distinguished, they can help the project team successfully gather and understand the requirements for the project. If the project team fails to distinguish the stakeholders at the beginning of a project, they will have to manage their requirements at a later stage, which could lead to project failure.

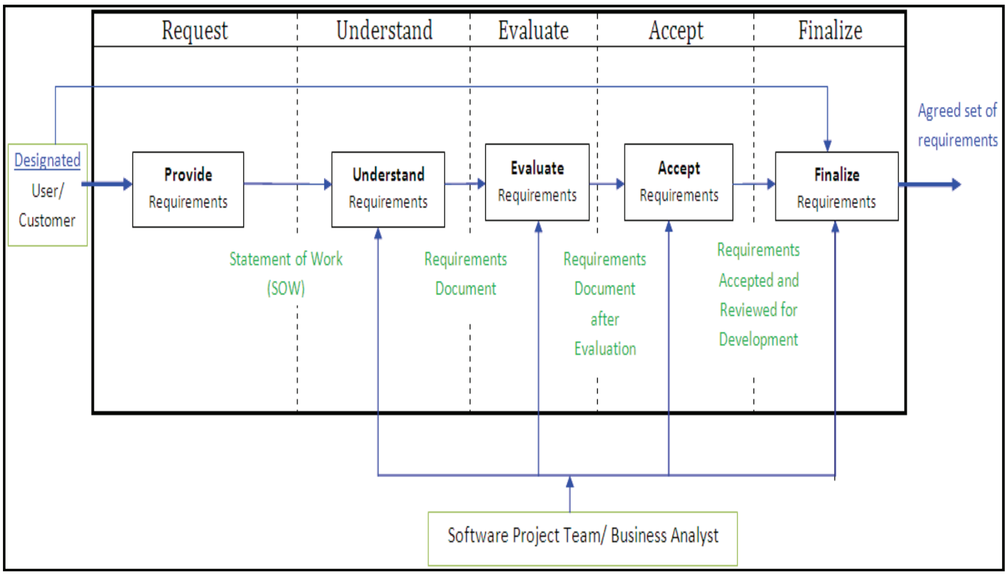

Figure 7: The proposed workflow model for SP1.1–"Understand requirements"

For distinguishing appropriate requirements providers, George Stonehouse et al. [59] highlighted significant criteria that can be used to distinguish between stakeholders. For example, a "primary stakeholder" is anyone in the organization who is vital, and an "active stakeholder" is any person who participates in the activities of the organization. Morris et al. [58] provided a list of criteria that helps identify and distinguish stakeholders based on influence, legitimacy, willingness to engage, contribution, and necessity of involvement. Schmeer [60] outlined various stakeholder characteristics, such as power, leadership, and knowledge, that can be used to describe the distinctive nature of the stakeholders. Based on cited references, Table 9 presents the proposed criteria the project teams need to setup and create for distinguishing appropriate requirements providers.

Table 9: Criteria for distinguishing appropriate requirements providers

No	Criteria	Question
1	Necessity of involvement	If this user/customer was not included, engaged, and involved during the development process, can he/she derail the process?
2	Contribution	Does this user/customer have useful information about the organization? Does this user/customer have knowledge about the product? Does this user/customer have the enough knowledge to contribute to the process?
3	Influence	Does this user/customer have decision making authority? Does this user/customer have strong power to manage decisions during the development process? Does this user/customer have the ability to assess impact of

		a change in the product?
4	Technical know	Does this user/customer know some technical issues about the product?
5	Level of commitment	Does this user/customer have strong level of commitment to the process and the product?

The second stage is "Understand." The understanding of the requirement begins with understanding the information contained in the SOW and stakeholder profile. Therefore, the input to this stage is the received requirements that are usually presented in different structures and forms (e.g., SOW) provided by designated users/customers. Based on the software application that needs to be developed, the requirements are categorized into different categories such as functional, technical, performance, and environment. In particular situations, some requirements may not be provided by the designated user/customer. In such cases, the project team must understand the business process flow of the designated user/customer in order to determine the missing requirements.

The third stage is "Evaluate." This stage has been included, because it is underlined as one of the basic sub-practices in CMMI v1.3 specifications [65] for SP 1.1. Furthermore, O'Regan [27] accentuated the significance of evaluating the requirements. The input at this stage is the generated requirements document after understanding the information obtained from SOW and the business process flow from the designated user/customer. In this stage, the business analyst of the software project team should interpret and clarify the requirements documents. In addition, reviewers are involved in order to judge the requirements against objective criteria, such as clarity, feasibility, consistency, and testability, for the evaluation of the requirements. This stage is essential, because it ensures that the requirements are reviewed and that no requirement is missed. The references cited in the literature emphasized the major role of reviewing the requirements in improving the understanding of the requirements (CMMI v1.3 specifications [65]; Chrissis et al. [17]; O'Regan [27]; Persse [64]; Kotonya and Sommerville [48]).

Under the section of SP 1.1, Chrissis et al. [17] and CMMI v1.3 specifications [65] provide examples of evaluation criteria. Also, O'Regan [27] suggests some criteria that can be used in the requirements evaluation. Based on these references, Table 10 presents the proposed criteria the project teams need to establish during planning for the evaluation of the requirements for development.

Table 10: Objective criteria for the evaluation and acceptance of requirements

Serial number	Criteria	Question
1	Clarity	Is each requirement properly stated? Does each requirement have only one interpretation?
2	Feasible	Can each requirement be implemented within available resources or technology?
3	Unique	Is each requirement uniquely defined with no duplication?
4	Prioritized	Is each requirement defined business for development?
5	Achievable	Can each requirement be achieved according to the budget and time constraints?

The fourth stage is "Accept." After the requirements have met all the objective criteria for the evaluation, the software project team will accept and acknowledge these requirements for development. It is necessary to point out that the evaluation and acceptance of the requirements will take place together. The "Accept" stage has been defined in our model, because the acceptance of requirements is explicitly indicated in CMMI v1.3 specifications [65], Chrissis et al. [17] and O'Regan [27] in the SP 1.1 section. It is also noted in the explanation of the Spiral model of the RE process (Kotonya and Sommerville [48]).

Lastly, "Finalize" is the last stage in the proposed model. In this stage, the requirements are finalized for development by directly involving the designated user/customer. After requirements pass the previous stages of "Request,""Understand,""Evaluate," and "Accept," the designated user/customer becomes involved in this final step. The reason for involving the designated user/customer in this stage is because it is stated in the fourth sub-practices for SP1.1 in CMMI v1.3 specifications [65] to "reach an understanding of requirements with requirements providers so that project participants can commit to them.""Finalize" is a mandatory stage in our model, because it will produce/output the agreed set of requirements. As stated in CMMI v1.3 specifications [65] for SP 1.1, "The result of these analyses and dialogs is a set of approved requirements." It is significant to underline that the agreed set of requirements is also known as base-lined requirements. After that, any changes can only be implemented through approval forms and the requirements change management process.

It is necessary to point out that guidelines have been provided to assist users in traversing the proposed model. Appendix A presents the guidelines including templates, forms, and checklists that can be utilized to traverse the proposed model. It is important to complete the mandatory information required in the template/checklists in order to complete a particular step before proceeding on to the next step.

5.2.2 SP 1.2–"Obtain commitment to requirements"

In this section, our findings, which include the guidelines and recommendations for obtaining the commitments to the requirements, are described. This section provides the answer to RQ1 for SP 1.2. Furthermore, suitable workflow models for obtaining commitments are provided. This section further illustrates the development of the proposed workflow model for SP1.2.

5.2.2.1 Related work on obtaining commitment to the requirements

Several research studies have been carried out in order to provide recommendations for better obtaining the commitments to the requirements:

- Verner *et al.* [52] indicated that the commitment to the requirements of the project participants can be obtained by requesting their participation in the development process.
- Niazi *et al.* [8] highlighted that if the project participants are involved and consulted in the software development process, they will be more confident. In addition, the authors further indicated that the setup of new software projects in the absence of consultation with all the stakeholders may lead to failure. The failure could occur because the project participants may have the impression that the system is not essential. Thus, they will not collaborate on efficient execution of the system.
- Persse [64] proposed five activities that can be used to implement SP 1.2:
    1. Identify appropriate approver groups: The main aim of this activity is to ensure that the people who should approve the requirements are clearly identified. These people may have been previously determined by the activity of identifying the reviewers. Occasionally, the group that is responsible for reviewing a document is also the one that approves it.
    2. Incorporate feedback: The basic idea behind this activity is the inclusion in the requirements sets of visible feedback from the reviewers obtained from SP 1.1. If relevant stakeholders feel that their

comments and feedback were given serious consideration, they will commit to the requirements and approve any change that might happen after obtaining the commitment.

3. Set a time limit: The project manager is fully responsible and accountable to keep the software development project running smoothly, according to the project schedule. The set timeframe for the project (with a reasonable deadline for the review process) is essential for obtaining commitment to the requirements.
4. Ensure that commitment allows for future change: The idea behind this activity is to ensure that people who will commit to the requirements have the chance to change the requirements as needed after obtaining the commitments. In other words, they can tweak and adjust the requirements through change control board approval and the requirements change management process.
5. Seek signatures: The form to obtain commitment to the requirements can be represented in many ways. The most traditional way is to seek signatures. Signing is a formal way to describe the commitment and the acceptance of the requirements.

- Hood [53] explained that one could obtain commitment to the requirements by approving the reviewed requirements during a project workshop.

- Michael [82] proposed a simple mapping of CMMI artifacts to SP 1.2, such as signing off on an agreed set of requirements and meeting minutes.
- O'Regan [27] stressed that all stakeholders and project participants must be involved in the process of committing to the requirements. This process encompasses all the requirements or changes that might happen after obtaining the commitment.
- Chrissis *et al*. [17] indicated the assessment of the impact of the requirements when they change is one of the primary sub-practices that needs to be performed in order to commit successfully to the requirements.

In addition to the above-mentioned studies, some models and process maps that seek to address and target the commitment process have also been proposed:

- Kasse [55] presented commitment process as a critical path for three process areas: requirements management, project planning, and project monitoring and controlling (see Figure 8). Core activities that need to be performed during this process, such as performing independent reviews of the plan and estimating the size, cost, and effort of the software project, are proposed. The author further indicated that in order to obtain commitment from the project participants, the differences between resource requirements and estimates must be resolved. In this case, stakeholders and developers should negotiate the differences in order to settle the conflicts successfully. All estimates must be negotiated and reconciled before all relevant participants can commit to them. The project teams, together with the supporting teams, must possess the belief that each requirement can be achieved based on the specified budget, time, and performance constraints. Collaboration with senior management is also required in order to review the recommitment and the internal or external factors.

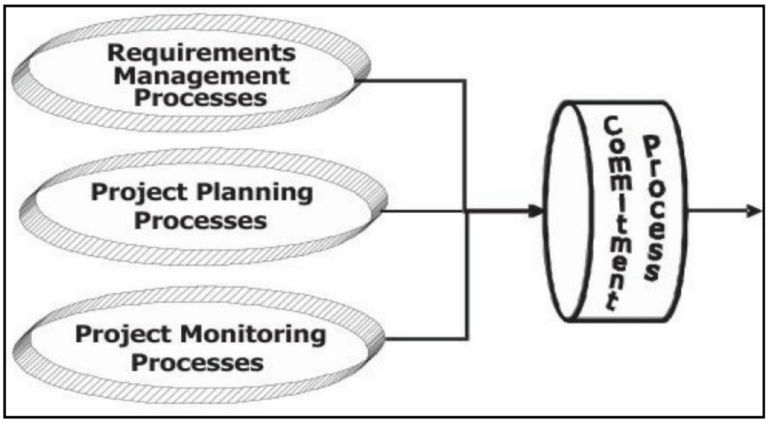

Figure 8 Commitment Process (Kasse, 2008)

- David Ing [56] proposed a workflow model for commitment management protocol (see Figure 9). The proposed workflow model for this protocol included important issues: (1) supplier and customer involvement in every stage of commitment, (2) defining new requirements through a request, (3) negotiation after receiving the request, (4) recording progress of the work and facilitating the assessment of the outcome, and (5) making the agreement closer to the report of completion. In this work, the author defined the term commitment as an agreement between two parties in a bid to produce a definite outcome they can accept if it meets the agreed conditions. In the context of the sense-and-respond system, the supplier is the first party, who is accountable and responsible for bringing the outcome. The other is the customer, who must approve the result when it satisfies his or her conditions. The author pointed out that defining a need or requirement will result in a request and will aid in negotiation and reconciliation. Agreement is also the cause of performing the required activities to achieve the outcome successfully. The reporting process is also vital, because it records the progress of the work and facilitates the assessment of the outcome. The author further emphasized that the system is susceptible to risk if a commitment is made directly after receiving a request and without negotiating and reconciling with the relevant project participants. The risk is also higher when feasibility studies are not carried out prior to obtaining the commitment. Therefore, negotiations should take place immediately after receiving the request in order to minimize the risk.

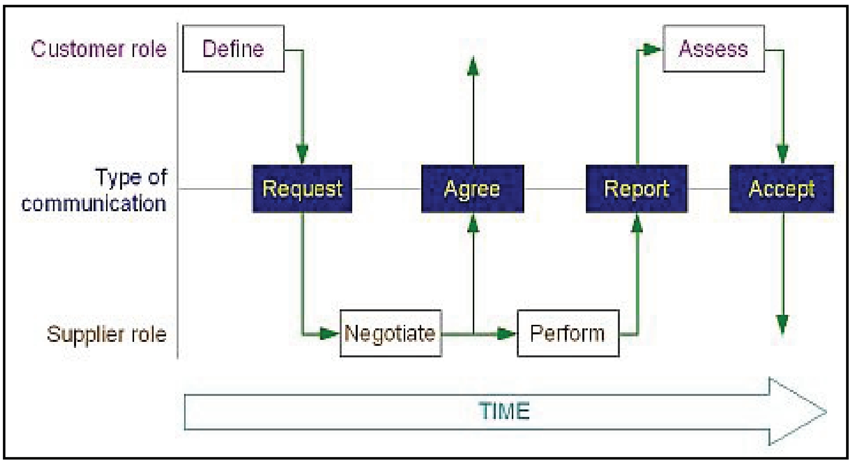

Figure 9: The Commitment Management Protocol (David Ing [56])

- Anum et al. [83] addressed the development process improvement in the context of the Services Software-as-a-Service (SaaS) cloud environment. The authors identified the main activities that need to be performed by organizations in order to achieve CMMI Level 2 certifications in their SaaS cloud environments. A workflow model for SP 1.2 was proposed in the context of the Services Software-as-a-Service (SaaS) cloud environment. This model highlighted the importance of examining, evaluating, and reviewing the final version of project proposals before signing the contract. It also indicated that the commitment to the requirements should be obtained by signing the financial proposal and the technical proposal. In addition, the proposed model underlines that the project manager (PM), VP of Engineering, business development representative (BDR), and cloud assessment officer (CAO) have significant roles to play in the development process. These responsibilities include evaluating and reviewing the project proposals. The review process should also include the scope of the project, timelines, cost, and deliverables. The commitment of the PM, BDR, CAO, and VP of Engineering must be obtained by signing the financial proposal and the technical proposal. The commitment to the technical part of the project proposal indicates that the project tasks will be performed and completed using the provided resources, the given timeframe, and so forth. On the other hand, the commitment to the financial part of the project proposal points out that the economic feasibility study is carried out to assess the viability of the given project within the estimated cost. The board of director's commitment must be provided using the project approval form. Finally, a kick-off meeting can be conducted to obtain the commitment to the requirements from the whole project team.

5.2.2.2 The proposed model for SP 1.2 –"Obtain commitment to requirements"

Our model for SP1.2"obtain commitment to requirements" is based on the findings from the literature presented in Section 4.2.2.1. The proposed model describes the process flow for obtaining a commitment to the requirements for a software project from the project's participants. It also demonstrates how the project's participants can commit to the current requirements and any likely changes as requirements evolve.

Figure 10 shows the proposed workflow model for SP 1.2 "obtain commitment to requirements. The model is divided into five essential stages: "Assess," "Report," "Negotiate," "Record," and "Commit." The specific activities associated with each of these stages have been clearly indicated. The relevant stakeholders should "sign off" on the existing requirements, such as baseline requirements/agreed-upon set of requirements. If a new request is received, the five stages are conducted.

The first stage is "Assess." This stage is based on one of the primary sub-practices in CMMI v1.3 specifications for SP 1.2. David Ing [56]considered it the main stage in his commitment management protocol. In addition, Anum et al. [83] highlighted the importance of assessing the impact of proposed changes to existing commitments. The authors emphasized the significant role of evaluating, analyzing, and documenting the impact of the proposed changes on current commitments, the work breakdown structure, project planning, and other work products. At this stage, the project team and business analyst should have an understanding of the nature of the new requirements and the effects of the proposed changes in the requirements to existing commitments. They then should classify whether it is a new requirement or a requirement change to an existing commitment. After that, the project team should carry out an evaluation of the new or changed requirements as well as an analysis of the impact. It is necessary for the project managers to employ their expert judgment during the decision-making process in this stage. The activities in the "Assess" stage can be considered preparatory exercises for the second "Report" stage.

During the "Report" stage, the project team and business analyst are given the task of preparing a full-impact assessment report. The documents examined should include the efforts that might be needed for the change, as indicated in the commitment process proposed by Kasse [55]. Furthermore, any additional resources that are necessary for the implementation of the new requirements or the proposed changes should be clearly declared. It is important to point out that both the "Assess" and "Report" stages can take place together. In other words, these stages can be combined and all of their activities carried out as a whole. It is essential to highlight that the "Negotiate,""Record," and "Commit" stages should immediately take place if the new requirements or the proposed changes to the existing requirements have no significant impact. This allows the stakeholders to sign off on the requirements without delay.

The third stage is "Negotiate," which was included because it is highlighted as one of the main sub-practices in the CMMI v1.3 specifications [65] for SP 1.2. In addition, O'Regan [27] highlighted the important role of the "Negotiate" stage. Furthermore, Kasse [55] indicated that stakeholders and developers should both negotiate the differences between the resource requirements and estimates in order to settle the conflicts successfully. The author further stressed that the agreement on the terms of commitment should be reached through negotiation.

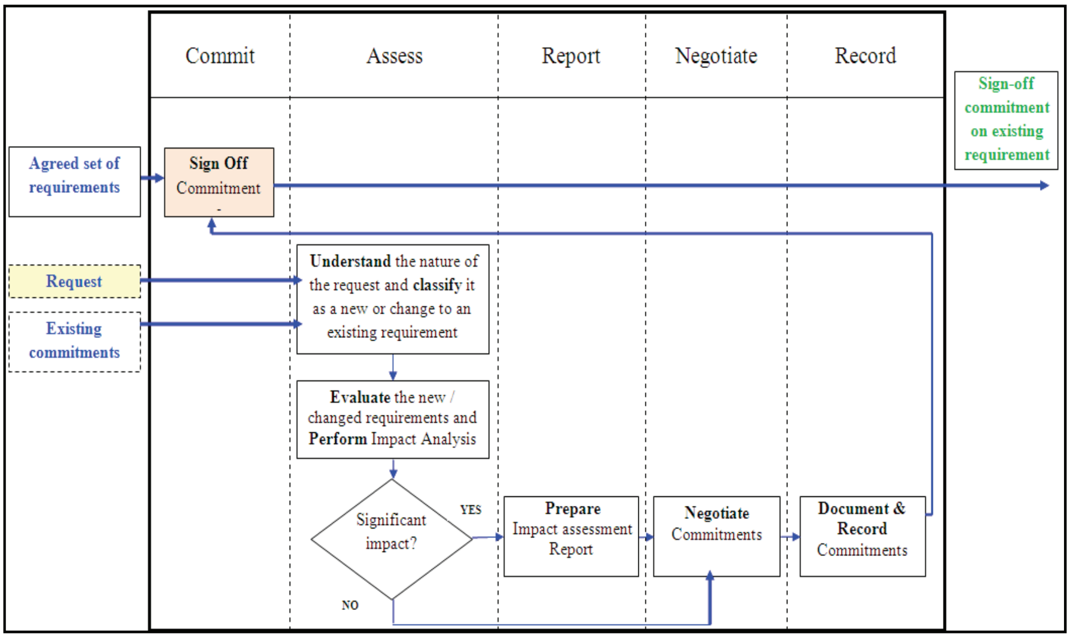

Figure 10: The proposed workflow model for SP1.2–"Obtain commitment to requirements"

Numerous research studies have emphasized the role of the negotiation stage in successfully obtaining the commitments from the participants on a project or the system's requirements (Chrissis et al.[17]; Ing [56]; Anum et al.[83]). Stakeholders and developers at this stage should negotiate and reconcile all the estimates until all the relevant participants can commit to them. Before moving ahead and implementing any of the new requirements or change requirements, the multiple viewpoints and variations between resource requirements and estimates must be resolved. The first step at this stage is to share the impact assessment report with relevant stakeholders so it can be reviewed. In other words, this report should be distributed among the stakeholders. This step can be performed through a formal discussion or by sending emails with impact assessment reports attached to the stakeholders and so forth. The second step is to give the stakeholders adequate time to review the report, so they can analyze the timelines, cost, and deliverables. The stakeholders are given opportunities to provide their comments, observations, and feedback after the review process. This can be achieved by conducting meetings, providing online feedback, sending emails, and issuing logs.

The fourth stage is "Record." This stage has been extensively defined in the model because the references cited in the literature stressed the significant role of the "Record" stage in successfully obtaining commitments to the requirements (CMMI v1.3 specifications[65]; Chrissis et al.[17]; O'Regan[27]; Anum et al.[83]). After the negotiation stage, the required documents and the impact assessment report should be updated. The second step is to prepare in full the necessary documents to be submitted to the stakeholders for the "sign off."

Finally, "Commit" is the last stage. This phase aims to win the commitment of the project stakeholders through a sign-off after they have reviewed the necessary documents. The project participants are allowed to provide comments and contributions on where the document may need further modification (Michael [82]).

This stage has been defined in the model, because it has been noted in the references that were cited in the literature (Moorthy [81]; Michael [82]). Appendix B provides guidelines, including templates, forms, and checklists that can be utilized to traverse the proposed model.

5.2.3 SP 1.3 "Requirements change management"

In this section, guidelines, recommendations, and suitable models that target requirements change management are identified from the literature. This section provides the answer to RQ1 for SP 1.3. Moreover, the development of the proposed workflow model for SP1.3 "requirements change management" is described.

5.2.3.1 Related work on managing requirements changes

Different research studies have been conducted to provide recommendations to effectively manage requirements changes:

- McGee *et al.* [67] and Pierce *et al.* [68] pointed out that the changes in the requirements are not problematic; the actual problem is how to deal with these changes in an effective manner.
- Nurmuliani *et al.* [70] pointed out that the first step toward effectively managing requirements change is identifying the sources of the change.

- Barry *et al.* [66] indicated that ambiguities in the initial set of requirements, as well as poor stakeholders' involvement in the project, drive the changes in requirements.
- Standish-Group [20] asserted that the management of the changes in requirements in an effective and efficient manner can be employed as an essential predictor of project success.
- Standish Group [72] indicated that requirements management and the change management process are important success points for the execution of small projects.
- Persse [64] presented five fundamental activities for implementing SP1.3:
    1. **Know that requirements will change:** This is indeed closer to a reminder than an activity. Change in the requirements is a likely element of the software development life cycle due to the dynamic nature of the software environment. Therefore, the project manager and developer team should be aware that the requirements will change during the project development lifecycle.
    2. **Control with baselines:** Data management is one of the most important activities in project management. It is important to identify and specify who requires the right to use which data at what time and with which privileges. A Requirements document (such as SRS) is one

example of the data that needs to be accessed. It is a technical document that moves through each phase of software development life cycle. Due to its high significance and wide-ranging impact, it should be managed with greater care and responsibility. The project manager and developer team are required to monitor changes in the requirements in order to ensure that they are working with the most up-to-date version of the set of requirements. Similarly, the modification requests will also be analyzed against the latest build.

3. **Honor your customers' needs:** The software project management team is accountable for two significant responsibilities. The first responsibility is to deliver a service or product that successfully fulfills the client's business needs. The next responsibility is to manage the amount of time, money, and resources that are allocated for the software project.
4. **Assess proposed changes:** One of the leading roles in project management is the assessment of the impact of the requirements whenever they change. All relevant stakeholders should be involved in this activity to evaluate the correctness and the needs of each change request. They have to identify what probable effects the opposed

change may bring to the project. The author recommends using a change review committee to assess the change requests.

5. **Incorporate changes in an orderly manner:** A proper mechanism has to be established in order to release the new versions of the baseline requirements. It is important to note that it is the project manager's work to communicate with relevant stakeholders and inform them about the result of the change impact analysis. The requirements document needs to be up to date in order to ensure that all team members have the latest version of this document. This will enable them to work on an updated set of requirements in a timely manner.

- Michael [82] proposed change audit report and Change Control Board (CCB) meeting minutes for SP 1.3.
- O'Regan [27] stressed that the change control should be conducted in order to assess the impact of the proposed changes.
- Chrissis *et al.* [17] emphasized that any change in the requirement should be documented and recorded. In addition, the change history should be maintained in order to trace the requirements volatility.

In addition to the above studies, some models and process maps that address requirements change management have also been identified:

- Spiral-Like [84] model consists of four core stages or phases: (1) problem owning, (2) problem solving, (3) system engineering and (4) technology-specific. In the first phase, a few modifications in the shape of latest features, addition or errors handling in the system are requested. In the closure of this cycle, the management team makes a decision on the importance of the modification and whether the change is required or not and how it will be implemented. The subsequent stage takes place when the proposed change has to be analyzed from a non-technical perspective. The third involves the preparation of the outline or layout for the implementation of the modification. Implementation is the final phase of the change request process. In this period, the requested change is implemented, and the verification of the technical solutions is performed. Although this model can be used for requirements change management in the software development lifecycle, it does not highlight any negotiation with the concerned customer.
- Olsen's [85] change management model is appropriate for software development as well as the maintenance phases. The users who propound the change in the requirement are regarded as the cause or source of the change. The proposed change is then moved forward to the change management phase. In this phase, change managers are entirely responsible for managing and controlling this change. Later, the accepted change is sent to the

implementation phase where the proposed change is integrated into the software project. The implementation of the approved change is then verified to ensure that the process is working correctly by testing the implementation code and carefully examining the documentation. After the verification phase, the change manager is responsible for issuing the change in the software project/system to their customers/users. It is important to note that, even though Olsen's model can adequately address the primary activities in requirements change management process, the model lacks some essential activities. These activities include batching, negotiation with the customer, understanding the change request and updating the documents. The activities mentioned above tend to be lost in this model. In other words, the Olsen's model does not accord proper attention to these activities. Moreover, the model does not provide any information about the actors in the change management process.

- Ince's [86] change management model supports basic functions, i.e. change request, reject, batch, implement, and update. In this model, the customers/users and the development team are the primary sources of change requests hence the change management process is initialized by them. As shown in Figure 11, the requested change is recorded and saved on the change request note. The change request note is then moved forward to the

Change Control Board (CCB) in order to make the decision about the change request. The CCB is the same for all change management processes regardless of change request type. In this proposed model, there are three typical types of decisions that can be made by the CCB. These include acknowledge/accept, reject/refuse, and batch. If the change request is recognized or accepted, the proposed change is then forwarded for implementation. Rejecting/refusing the change request implies that the proposed change cannot be applied and implemented hence cannot be incorporated and integrated into the software project or system. Batching the change request implies that the proposed change will occur after some period. A change authorization note must be completed once the change request is acknowledged and admitted for implementation. The proposed changes are then implemented from that point forward. The requirements document should also be consistently updated and modified, and the validation process should be performed immediately after implementation of the changes. Later, records of the validation and test activities should be created to report and file all changes that have occurred and integrated into the software project. Lastly, the configuration documents should be revised, and the stakeholders involved informed and notified about the implemented change. In summary, Ince's model supports the following basis functions: change request, reject, batch, implement, and update. However, it does not support the verification/confirmation activity. As a result, it is quite difficult to determine whether or not the implementation of the proposed change is working properly.

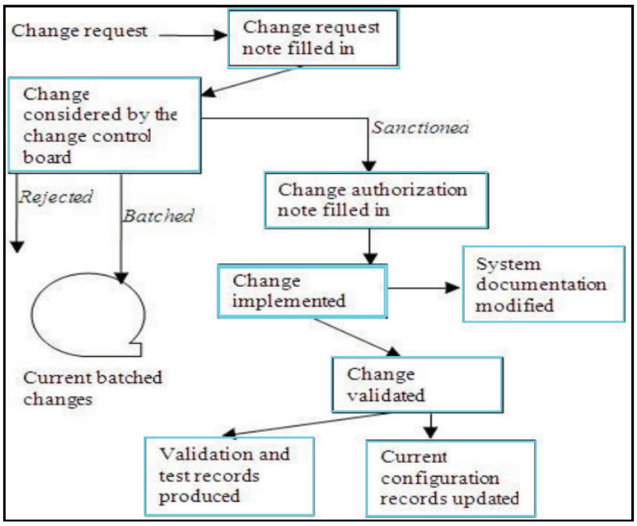

Figure 11: Ince's change process model (Ince, [86])

.

- Niazi et al. [87] introduced the requirements change management (RCM) model for implementing SP 1.3. The RCM supports software development organizations in the change management process. This model has five core stages: request, validate, implement, verify, and update (see)The research was based on two data collection sources. The first source is an extensive literature review of SPI and RE that includes research articles, published experience reports, and case studies. This is performed in order to provide a list of characteristics that effectively help in managing the requirements change process. Three RCM models are identified for implementing specific practice Level 2 of CMMI. These RCM models are Olsen's change management model, Spiral-Like change management process, and Ince's change process model. The second source involves interviewing experts in SPI from two companies and aims to identify the major interest of these representatives in their RCM processes. The researchers used "ease of use" and "user satisfaction" as essential criteria for building the RCM. The first stage is "Request" whereby the relevant stakeholders (either internal or external) to the software project initiate RCM by requesting a change. A requirements change pool is used to record and store the primary information about the change request such as a clear illustration of the proposed change, justification of change, and the people who initiated the change request. The second stage is "Validate" and it involves getting a clear understanding of the proposed change. After this stage, analysis of the proposed change should be performed in order to determine the impact of the change in terms of the cost, schedule, effort, and risk. The next phase involves making a decision on whether or not the change request is accepted. If the change in the requirements is accepted, then it is passed to the "implement" stage in order to implement and integrate the approved changes to the software project.

“Verify” is the fourth stage in RCM and it entails checking whether or not the implementation of the recommended change is working correctly. If the verification phase is not satisfactory, the change request is carried back to the “Validate” stage for additional understanding, observation and evaluation of the change request. Lastly, the requirements document is updated, and the customers/users informed about the implemented change. The final version of the requirements document is then distributed to the concerned stakeholders. Lastly, the software products are released to the consumers. It is important to highlight that the presented RCM is validated by performing the evaluation using the expert review process.

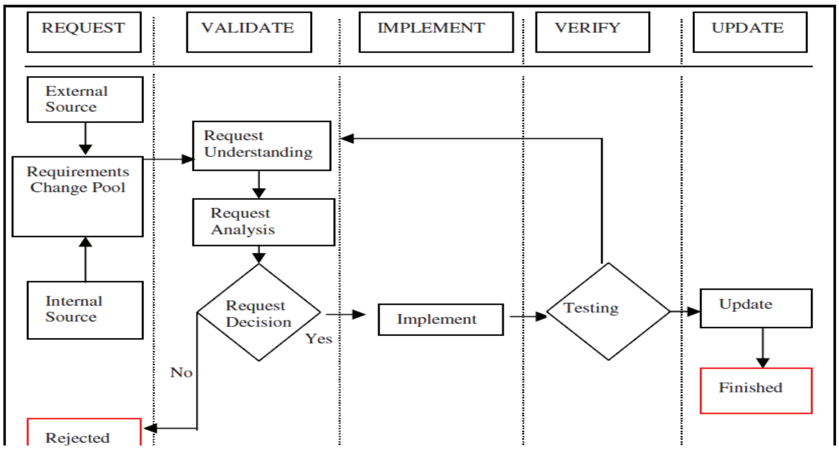

Figure 12: Requirements change management model (Niazi et al., [87])

- Bhatti et al. [88] proposed a methodology to deal with the changes in the requirements of a software project. The proposed methodology consists of six basic phases: initiate, receive, evaluate, approve or disapprove, implement, and configure the change requests. In the first phase (initiate), the relevant stakeholders of the software project begin the change management process by requesting a change. "Change Request" is a typical work product that can be used in this phase. The second phase is "receive" whereby the request for change is received in a particular form that follows the standard for seeking a change. The form is then prepared for consideration and evaluation. The third phase is "evaluate" in which the impact of the proposed change is assessed and analyzed based on the current commitments and project plans. Timelines and cost analysis of the proposed changes should also be included in this phase. It is highly recommended that quality assurance teams be involved in the evaluation and the analysis. An “Evaluation Sheet”, which shows the evaluation of the proposed changes in detail, can be used as a typical work product in this phase. The decision of approving or disapproving the requested change is mainly based on the possibility and practicability of the proposed changes. If the requested change is admitted and accepted, it is sent forward to the next phase, i.e., “implement.”In this phase, an “approval notification form” can be utilized as typical work product. If not, the person who initiates the change request is informed about reasons for rejection through the “reasons of disapproval form.”In the “implement” phase, the approved changes are incorporated and integrated into the software project. Last, the “configure” stage entails maintaining and storing all change requests in a configuration management repository. All activities associated with this period should be carried out by a configuration engineer. A “List of configuration items” can be utilized in this phase as a classic work product. It is necessary to point out that this methodology lacks some vital activities such as verification and batch. Therefore, it may not indicate whether or not the implementation of the approved changes is working satisfactorily.

- Anum et al. [83] proposed a workflow model for SP 1.3 for SaaS cloud environment. In this workflow model, the relevant client of the software project/system initiated the workflow model for change management by requesting a change to the project manager (PM) and the business development representative (BDR). In case the change request was submitted to BDR, it was forwarded to PM. A repository was used to store and record the change request. The PM and CCB should be involved and participate in evaluating the change request. The impact of the change was evaluated based on the current commitments, work breakdown structure, project plan, and other work products. The authors defined a number "n" for the impact of change. If the impact did not exceed "n", the proposed changes were accepted immediately. If not, a review meeting took place, in order to make a decision on whether or not to approve the proposed change. The client was informed of the evaluation results and the decision of approving or disapproving the change request. The business development representative (BDR) was responsible for informing the relevant client about the results and the decision taken regarding the change request. In case the proposed change was accepted, the requirements document, as well as the project plans must have been well updated and maintained.

5.2.3.2 The proposed model for SP 1.3 "Requirements change management"

In this section, we propose an implementation model for the CMMI Level 2 specific practice SP 1.3 "manage requirements changes" in the REQM PA. The proposed model is based on our findings from the extensive literature review presented in Section 5.1.1. It effectively controls and deals with any change in the requirement that may happen after obtaining the commitment. It has six essential stages: "Initiate," "Validate," "Implement," "Verify," "Update," and "Release." Particular activities associated with each of these stages have been clearly pointed out. Figure 13 shows the proposed model for SP 1.3.

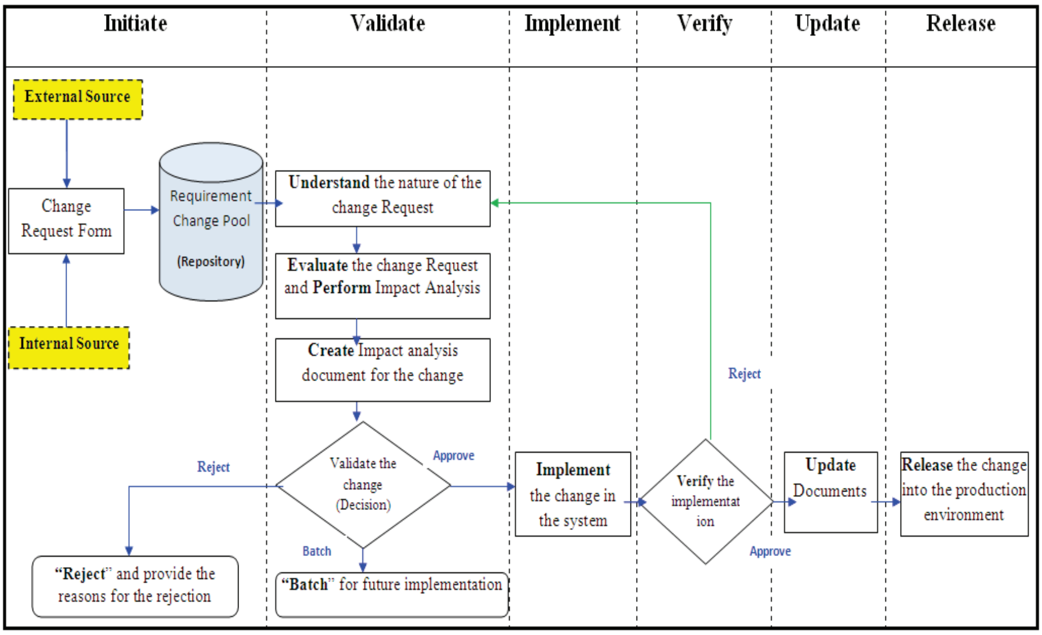

Figure 13: The proposed workflow model for SP1.3–"Requirements change management"

The first stage is called "Initiate." This stage is incorporated in the model, and it has been highlighted in the following requirements change management models: Ince [86], Niazi et al. [87], Bhatti et al. [88], and Anum et al. [83]. In this stage, the relevant stakeholders identify the need for change in the requirements. Internal and external project stakeholders can initiate the requirements change model. The initiation is implemented by requesting a change by filling out a change request form. The change request is then fed to a requirements change pool (repository). A requirements change pool is used to record primary information about the change request, such as justification of the change and the people who initiated the change request. It is necessary to indicate that the repository is included in our model because the requirements database is considered as one of the essential work products in CMMI v1.3 specifications [65] for SP1.3.

The second stage is "Validate." This stage is included in the model as highlighted in the following references cited in literature: Spiral-Like change management [84]; Ince's change process model [86]; Niazi et al. [87]; Anum et al.[83]. This stage involves acquiring a clear understanding of the change request. The nature of the new requirements or the proposed changes in the requirements of existing commitments should be well understood. An evaluation and an analysis of the change request should then be carried out to determine the impact of the change in terms of the cost, schedule, effort, and risk. It is important to point out that the evaluation activity of the impact of the change is included because it is indicated as one of the main sub-practices in CMMI v1.3 specifications [65] for SP1.3. Later, a full-impact assessment report should be prepared. This report should clearly show the required cost, updated schedule, and any additional resources necessary for the implementation of the change request. The final activity in this stage is decision making. This proposed model contains three types of decisions: approve, reject, and batch. If the change request is approved, it is forwarded for implementation. Rejecting the change request implies that the proposed change cannot be applied, hence cannot be integrated into the software project. Batching the change request suggests that the proposed change will occur after some specified period (future implementation).A change can be rejected due to lack of clarity or inadequate budgets. A change could be batched when it is observed to be of much lower priority in the current scenario but can affect the system at a later time.

The third stage is "Implement." This stage is defined in our proposed model because it is pointed out in the following references cited in literature: Spiral-Like change management [84]; Olsen's change management model [85]; Ince's change process model [86]; Niazi et al. [87]; Bhatti et al. [88]; Anum et al. [83]. If the change is accepted, it is passed to the "Implement" stage, where the approved changes are integrated into the software project. This also involves all the required changes to the affected work products, such as design, code, and test cases.

The fourth stage is the "Verify" stage. This step is included in order to check whether the implementation of the recommended change is working correctly. This stage is added because it is described as one of the primary steps in the following proposed models: Olsen's change management model [85] and Niazi et al. [87]. In this stage, the implemented change is verified for completeness and correctness before being moved to the next stage. It is important to indicate that the change request is carried back to the "Validate" stage for purposes of additional understanding and evaluation if the verification phase is not satisfactory. This stage also helps ensure that there is no negative impact on the previously working modules.

The requirements documents are then "Updated." The work product revision history is updated to reflect the latest change(s), and the revised work products are updated into the system repository. Furthermore, the customer/user is informed about the implemented change. The final version of the requirements document is distributed to the concerned stakeholders. This stage is incorporated because the following references cited in the literature stress the significant role of updating the requirements for managing the change in the requirements: O'Regan [27]; CMMI v1.3 specifications [65]; Chrissis et al. [17]; Persse [64]; Olsen's change management model [85]; Niazi et al. [87]; Michael [82]; Anum et al. [83]. Moreover, CMMI v1.3 specifications [65] underline that documenting the requirements change, maintaining the requirements change history, and making the change data available for the project are basic sub-practices for SP1.3.

To conclude, "Release" is the final stage. In this stage, the change is released into the production environment, and the final software products are released to the consumer. It is necessary to indicate that we have provided guidelines to help the users in traversing through the proposed model for SP 1.3. This model is broken down into simplified steps. Each of these steps is linked to corresponding templates/checklists. It is important to complete the mandatory information required in the template/checklists in order to complete a particular step before proceeding on to the next step. Appendix C presents the guidelines including templates, forms, and checklists that can be utilized by the user to traverse the proposed model.

5.2.4 SP 1.4 "Maintaining bidirectional traceability of requirements"

This section describes our obtained results from the literature, including the guidelines and recommendations for maintaining and supporting the bidirectional traceability of requirements. Suitable models that would help trace all lower-level requirements to a valid source and vice versa are also identified. This section provides the answer to RQ1 for SP 1.4. Moreover, this section describes the development of the proposed model for SP1.4"maintaining bidirectional traceability of requirements."

5.2.4.1 Related work on maintaining bidirectional traceability of requirements

Several research studies have been carried out to provide recommendations to better maintain requirements traceability:

- Gorschek [75] and Gorschek *et al.* [76] underlined that requirements traceability is one of the key activities of requirements management, which is an essential part of RE.
- Ramesh [77] highlighted that requirements traceability assists in identifying the source for each requirement and the issuer.

- Abran *et al.* [79] indicated that requirements traceability helps in managing changes that may happen after obtaining commitment to the existing requirement. In other words, traceability is required during the impact analysis activities in order to evaluate the effect of each requirement due to the proposed change.
- Persse [64] stressed that there is no need to have a certain tool to perform SP1.4. Simple spreadsheets could be used to maintain the traceability of the requirements.
- Moorthy [81] considered traceability matrix to be a main CMMI artifact for SP1.4.
- Michael [82] considered Requirement Traceability Matrix (RTM), project review meeting minutes, CCB meeting minutes, and project status reports to be main CMMI artifacts for SP 1.4.
- O'Regan [27] indicated that SP 1.4 can be achieved by using RTM as a typical way to trace all lower-level requirements to a valid source and vice versa. RTM can be implemented by using a simple Excel spreadsheet.
- Chrissis *et al.* [17] pointed out that generating RTM is a necessary sub-practice in order to maintain the bidirectional traceability of the requirements.
- Anum *et al.* [83] indicated that RTM should be generated in order to perform the traceability. This matrix is a two-dimensional array that lists all

requirements of the software project in one column. The remaining columns are used to list project related activities, plans, and other work products.

Furthermore, some models and process maps that address the requirements traceability have also been identified from the literature:

- Ramesh et al. [89] proposed a low-end traceability model based on multiple empirical studies. This model is appropriate in the applications that involve the following operations on the requirements: decomposition, allocation, compliance verification, and change control. Software systems that can adopt such a model should have the typical complexity with about 1000 requirements. In this model, requirements traceability could be seen as a link between the initial requirements and the real components. High-level system requirements were used in order to derive and obtain low-level requirements. The original requirements, as well as lower level refined requirements were derived from higher level system requirements. The Original and derived requirements were allocated to the system and its components. All the requirements were verified by determining the dependencies between requirements and system components. In other words, capturing the components that satisfied requirements and mapping components to requirements was necessary in order to verify that all requirements were perfectly tackled. For the compliance verification phase, the requirements database was used to retrieve the last updated version of the system’s requirements. After that, tests or simulations were developed. The proposed model highlighted that once any change is triggered, the traceability links are used to determine the compliance verification procedures. In addition, a requirements database should be used in this model to retrieve the last updated version of the system's requirements. The obtained results from the tests or simulations were used to show that all of the requirements were well addressed and met

- Kirova et al. [90] proposed a traceability model for Alcatel-Lucent's Wireless Business Group project. The model is divided into four stages: "customer requests,"" features," "requirements and architecture artifacts," and "development and verification artifacts". In this model, a typical tracing starts with features provided through customer requests and goes through requirements levels, including architecture, design, and test. The proposed model provided the opportunity to create the multi-dimensional traceability matrix in a gradual manner. It is important to underline that the bidirectional traceability of requirements were maintained at each level of the decomposition. Furthermore, “system requirements” as well as “architecture requirements” were mapped to high-level and/or low-level design. They were also linked to verification test cases. To support this model, the authors implemented automated traceability environment (also known as TraceabilityWeb). TraceabilityWeb strongly helped investigate repositories that were needed for the artifacts. It also assisted in auto-generating the maps between the artifacts. Azmi et al. [91] highlighted that the traceability model proposed by Kirova et al. [90] had the ability to trace the artifacts at different levels of granularity. In addition, it was tool supported. Furthermore, both configurable control and notification procedures were adopted in this model. The author also underlined that the change impact analysis process that was adopted in this model was simple and smooth. On the other hand, such model was only applicable and appropriate to small teams.

5.2.4.2 The proposed model for SP 1.4 "maintaining bidirectional traceability of requirements"

Our model for SP1.4 "maintaining bidirectional traceability of requirements" is mainly proposed based on our findings from the literature presented in Section 4.2.4.1. This section illustrates our findings, including the guidelines, recommendations, and suitable models for maintaining the bidirectional traceability of requirements. Figure 14 shows the proposed workflow model for SP 1.4. The proposed model is divided into six essential stages: "Request," "Maintain," "Validate," "Allocate," "Verify," and "Release." In addition, we clearly presented certain activities associated with each of these stages.

Our model starts with a relevant user/customer providing the new/changed requirements request. Then, the request is fed to the requirement/feature (repository) in order to record basic information about the request. This stage is included in our requirements traceability model, as it was highlighted in the traceability model proposed by Kirova et al. [90]. It is necessary to underline that such a repository is included in our requirements traceability model because CMMI v1.3 specifications [65] for SP1.4 highlighted that the traceability can be performed by using simple spreadsheets, databases, or any other ordinary tools. In addition, requirement/feature (repository) has been cited in the traceability model that was introduced by Kirova et al. [90]. It is necessary to point out that CMMI v1.3 specifications [65] stress the important role of SP1.4, especially in the case of assessing the impacts of the requirements changes on the current commitments, work breakdown structure, project plan, and other work products.

The second stage in our model is "Maintain." This stage involves developing a clear understanding of the request. The analysis activity is needed because the user requirements are predominantly high-level, thus, they may also need to be split into more manageable parts. In addition, any implied and derived requirements should also be identified and recorded into the software requirements specifications. This includes mapping all lower-level requirements to a valid source and vice versa. This stage is included in the model, as highlighted in the following references cited in literature: CMMI v1.3 specifications [65]; Chrissis et al. [17]; O'Regan[27]; O'Regan [27]; Abran et al. [79]; Anum et al. [83].

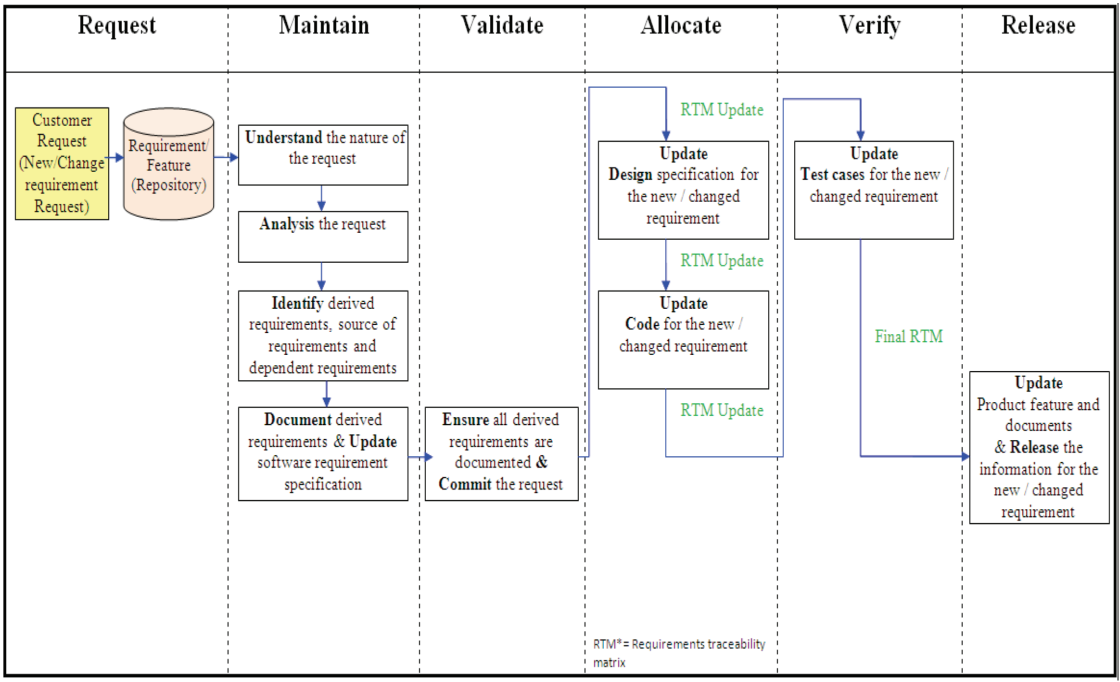

Figure 14: The proposed workflow model for SP1.4–"maintaining bidirectional traceability of requirements"

The third stage is "Validate." This stage is needed to review the updated software requirements specifications in order to ensure that all the derived requirements are well-mapped and documented. After that, relevant stakeholders commit the request as well as its impact (Persse [64]). All the signed-off requirements are updated into the RTM.

The fourth stage is the "Allocate" stage. The requirements are allocated to the work products across phases based on the impact of the requirements changes. The derived requirements are then translated into the design of the end product. As a result, design, code, and the corresponding sections in RTM specifications are updated. O'Regan [27], Moorthy [81] and Michael [82] highlighted that the requirements traceability matrix must be updated after any change that may happen after obtaining commitment. Therefore, the requirements traceability matrix should be updated at the end of each phase in order to depict the linkage of each of the requirements.

The fifth stage is the "Verify" stage. This stage is included in order to check whether the implementation of the finished product is working correctly. Test cases for the finished product are created to ensure that all the requirements are well addressed and met. This stage is added because it is described as one of the primary activities in the following proposed models: Low-End Traceability Model (Ramesh et al. [89]) and traceability model (Kirova et al. [90]). In this stage, the final product, including the proposed change, is verified for completeness and correctness. Once the test cases are created, they are updated into the RTM. In case of any further change in test cases, which might have an impact on other modules, the RTM must be updated accordingly. The latest update on design, code, test cases, the final RTM, and the final work products should be recorded into the system repository.

The final stage is "Release."In this stage, the final software product is released to the consumer. In addition, product features and requirements documents are updated. Appendix D provides guidelines including templates, forms, and checklists that can be utilized by the user to traverse the proposed model.

5.2.5 SP 1.5 " Ensure alignment between project work and requirements"

This section presents our findings from an extensive literature review of SPI and RE, SPI models such as CMMI, and software requirements engineering. Guidelines for ensuring the alignment between the project plans and the work products and requirements are identified. Moreover, suitable workflow models that would help keep the requirements updated and synchronized with the project work products are illustrated. This section further presents the development of the proposed workflow model for SP1.5“Ensure alignment between project work and requirements.”

5.2.5.1 Related work on ensuring alignment between project work and requirements

Various studies have been done to adequately ensure alignment between project work and requirements:

- Jonasson [92] pointed out that identifying inconsistencies between the project work products and the requirements is one of the key activities of requirement management, which is essentially part of RE.

- Persse [64] indicated that the SP1.5 requirements could be met by holding periodical meetings, conducting peer-reviewed examinations, and performing in-progress checkings. The author further highlighted that one of the most significant responsibilities of project management is to ensure that the software project plans perfectly reflect the state of the requirements at any point in time. To ensure this alignment, the requirements should be compared with project milestone deliverables in prearranged and established phases. Review meetings can also be held to achieve this practice.
- Moorthy [81] proposed “updating of project plan” to be a typical CMMI artifact that could be used by organizations to satisfy SP1.5.
- Michael [82] considered peer review defect logs and quality assurance audit reports as core artifacts for implementing SP1.5.
- O’Regan[27] indicated that the requirements management process area ensured that the requirements were kept aligned with the software project deliverables. Throughout the project execution, the software project management should maintain the latest version of the approved set of requirements. From this set, all software project activities should be accurately planned. Therefore, SP1.5 should be introduced as specific practice in the requirements management process areas.

- Chrissis et al. [17] stressed that "indicating the main cause of the inconsistency" and "identifying the changes needed to be performed on the project plans and work product" are regarded as basic subpractices that satisfied SP1.5. The authors further indicated that corrective actions should also be established to resolve inconsistencies. In addition, the requirements traceability matrix (RTM) is highlighted as a typical CMMI artifact that could be used to help in reviewing process.

In addition to the above research studies, some models and process maps that target the alignment between project work and requirements have also been proposed:

- Kasse [55] indicated that once requirement change request was approved, all life-cycle work products should be kept consistent, and the software requirements specification should be updated accordingly. The review process should also be performed on all related project work products to identify any change that has taken place. This was done to ensure the consistency and integrity of the project work products and the requirements. It is, therefore, necessary to maintain the consistency of the design, coding, and testing artifacts with the requirements throughout the lifecycle of the software project. Thus, the relationships between the requirements of the software project, design, coding, and testing artifacts should be accurately traced. The changes that may affect the requirements should also be tracked. In addition, the inconsistencies between the software artifacts and the requirements of the project should be clearly identified, and action items should be created in order to resolve the inconsistencies. In terms of configuration auditing, the functionalities of the software product should be compared to the requirements. Furthermore, the documentation that includes maintenance activities should also be aligned with the requirements. Software product components should satisfy the software requirements as well as all the accepted proposed changes in the requirements. Examination and evaluation need to be performed in order to verify that the software product and its work products and documentation are consistent. In order to check and verify the consistency of the software product specifications, requirements traceability matrix can be used.

- Anum et al. [83] proposed a model for CMMI Level 2 specific practice 1.5 in the context of Software-as-a-Service (SaaS) cloud environment. In this model, the reviewing process is conducted through project activities, plans, and project work products in order to keep them consistent with the requirements as well as with the requirements changes. The authors also highlighted that if any inconsistency is triggered, corrective actions should be put in place in order to resolve it.

5.2.5.2 The proposed model for SP 1.5 " Ensure alignment between project work and requirements"

The model for SP1.5“Ensure alignment between project work and requirements” has been developed. It has been proposed based on the findings from the literature presented in Section 5.2.5.1. This proposed model effectively keeps the requirements updated and synchronized with the project work products. Six core stages, namely, request, plan, review, identify, discuss, and rework, have been included in the model. In addition, we have described certain activities that are associated with each of these stages (see Figure 15). The model begins with a request to review activity at every phase of the development of the software. This can be initiated as well when there is either a key change made or a group of important changes that have to be carried out in the project.

The second stage is “plan.” This stage involves developing a clear understanding of the request. Following that, the project manager and business analyst need to create a review plan that helps conduct the review process of the artifact. During this stage, the project manager needs to make sure the plans and work products are both fully ready and available for the review process. In addition, the availability of the resources as well as the review procedures that are required to conduct a full technical review should be guaranteed and confirmed. It is also possible he/she might occasionally adjust the schedule so as to match the resource’s availability.

"Review" is the third stage, where a technical review of the artifact, RTM, work products, and project plans is undertaken by the reviewers. This stage also involves review of the design, code, and test cases. It is defined in our model because it is noted in Chrissis et al. [17], Persse [64], and Anum et al. [83]. It is also highlighted in the form of CMMI mapping for REQM PA at CMMI Level 2 that was proposed by Michael [82]. The review stage is considered a critical stage because an effective and efficient review process will result in ensuring the alignment between project work and the requirements. This stage also helps to identify any inconsistencies that exist between the various work products. RTM is underlined by Chrissis et al. [17] as a typical CMMI artifact that can be used to help in the review stage. Furthermore, Kasse [55] and Anum et al. [83] highlighted the important role that RTM plays in making sure that the project work and the requirements are aligned correctly.

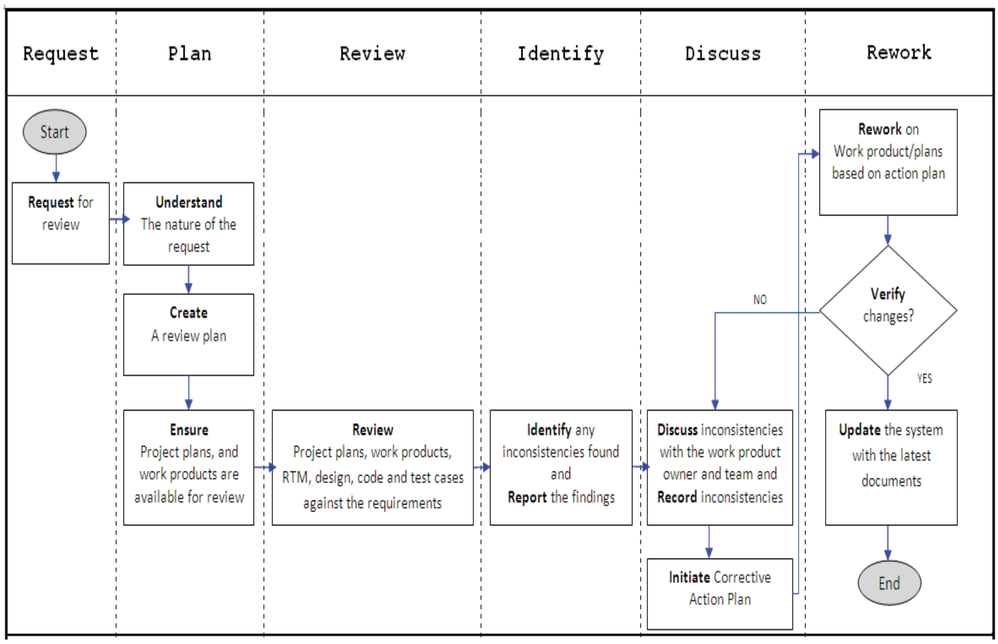

Figure 15: The proposed workflow model for SP1.5–"Ensure alignment between project work and requirements

"Identify" is the fourth stage. This is when the reviewers highlight their findings, list them correctly, and indicate any inconsistencies they see in the artifact. CMMI v1.3 specifications [65]stress how important it is to identify and indicate the sources of such inconsistencies. As a result, this stage is incorporated into our model. Also, this stage is cited in the model proposed by Anum et al. [83] for identification of inconsistencies in SaaS cloud environment.

The fifth stage is "discuss." This stage is introduced in our model as noted in the following references cited in the literature (Persse [64]; Anum et al. [83]). This stage is included so as to share the findings with the owner of the work product and the project team, including the project manager. An analysis of the findings is then carried out. This is because it is possible a reviewer could have failed to understand the design or requirements of the project. A review feedback document is then used to record the findings that have been validated. Based on the findings that indicate the verified inconsistencies, corrective actions are initiated in order to resolve these inconsistencies. Any rework that is performed on the work products/plans will be based on these corrective actions. It is important to highlight here that initiating corrective action is a primary subpractice that is proposed by the CMMI v1.3 specifications [65] for SP1.5. Moreover, corrective actions are regarded by Chrissis et al. [17], O'Regan [27], and Anum et al. [83] as being essential in resolving any inconsistencies that have been found between the project plans, the work products, and the requirements.

Finally, "rework" is the last stage in the proposed model. In this stage, the rework process is performed on the work products or project plans based on the initiated corrective actions. This stage is included in the our model because in CMMI v1.3 specifications [65], Chrissis et al. [17], Kasse [55], O'Regan [27], and Anum et al. [83] underlined the importance of applying the corrective actions on the work products/plans to resolve the inconsistencies between the project work and the requirements. This implies the strong need for having the "rework" stage, which performs rework process on the work products/plans based on the initiated and agreed corrective actions.

The necessary changes of the work products or project plans are then presented for a verification process to ensure that no feedback is missed and that the project work is aligned with customer requirements. Once the documents are verified, all review findings are incorporated, and the project work is aligned with the requirements, and the work products and all associated documents are updated into the system. If the verification is not successful, all related documents are then sent back to the "discuss" stage for further analysis of the inconsistencies between the project plans, the work products, and the requirements.

5.3 Evaluation of the proposed models

5.3.1 Evaluation of the proposed models through an expert panel review process

We conducted an expert panel review process to seek the opinions of software process improvement experts about different aspects of the evaluation of our proposed workflow models. It is vital to mention that all expert reviewers involved in the evaluation process have adequate knowledge of SPI implementation, SPI models (such as CMMI), and software requirements engineering. As far as identifying the expert reviewers is concerned, it is worth mentioning that one of the expert reviewers was an acquaintance through whom we were able to reach many other expert reviewers. The SPI reviewers involved in the evaluation were selected based on their experience in the field of software process improvement. The experts' profiles are presented in Table 11.

Table 11 SPI experts’ profile

SPI expert	Job title	Experience of SPI expert (years)	Knowledge of CMMI (Low 1 - 5 High)
1	Project manager	15	4
2	Project manager	13	4
3	Project manager	9	4
4	Software developer	7	3
5	Software developer	10	4

The SPI experts were categorized into three categories based on their experience and knowledge. According to Khan et al. [133], the researchers can define their own criteria in order to determine different levels of expert reviewers.

Table 12: CATEGORIES OF EXPERTS

Experts’ level	Experience of SPI expert (years)	Knowledge of CMMI (Low 1 - 5 High)
Junior	(1–8) years	1-3
Intermediate	(9–12) years	4
Senior	(13+) years	4-5

In order to obtain SPI experts' feedback about our proposed models, a questionnaire was developed. Some questions were adapted from (Niazi [93]) to satisfy the objectives of this research. The questionnaire consists of three core parts: a cover letter (consent form) in which we briefly describe the purpose of this evaluation, demographics, and model feedback. The model feedback portion addresses the following important aspects: practice satisfaction, ease of use, user satisfaction, and applicability to small- and medium-sized software development organizations. Before asking the SPI experts to fill out this questionnaire, an academic researcher reviewed and examined the questions. Based on his feedback and comments, we have made changes to some questions in order to make the questionnaire clearer and more logical to respondents (see Appendix E). A summary of the main conclusions from the data collected is presented below and shown in Table 13.

In regards to the question related to "practice satisfaction," on a five-point scale where the choices range from "strongly agree" to "strongly disagree," all the expert reviewers selected either "strongly agree" or "agree" for both models that were proposed for SP 1.1 and SP1.2. In other words, for the proposed model for SP 1.1, three of the experts selected "strongly agree," while the other two selected "agree."For the model proposed for SP1.2, one of the experts selected "strongly agree," and the other four selected "agree." For the proposed models for SP1.3 and

SP 1.4, four out of five experts chose either "strongly agree" or "agree" for each of them. For one proposed for SP 1.5, four out of five experts chose “agree.” These results clearly indicate that the proposed models satisfied the CMMI v1.3 specifications, as the question related to practice satisfaction was designed according to the objectives of each specific practice stated in the CMMI v1.3 specifications.

Two clearly defined questions were asked about "ease of learning and ease of use" (i.e., RQ2) in the evaluation form for each specific practice. The first question was related to the overall clarity of the proposed model's representation, and the second was about the range of knowledge that is required to use the proposed model. In other words, the experts were asked the following two questions about "ease of learning and ease of use" (i.e., RQ2) for each proposed model:

1. How clear is the representation of the proposed workflow model using a scale of 1 to 5, with 5 = "very clear" and 1 = "not at all?"

2. How much knowledge of CMMI is required to learn how to use our proposed workflow model using a scale of 1 to 5, with 5 = "too much knowledge" and 1 = "not at all?"

For the first question, four of the five experts chose either 5 or 4 on a five-point scale that ranged from "very clear" to "not at all" for the proposed model for SP 1.1. For the model proposed for SP 1.2, three of the expert reviewers selected 5, while the other two experts selected 4 on the same scale. For the one proposed model for

SP1.3, all of the experts selected "5" or "4" on the five-point scale for the proposed model for SP1.3. For the proposed model for SP1.4, all of the experts selected "5" or "4" on the five-point scale, except for one expert who selected "3." For the one proposed for SP 1.5, four out of five experts chose “4”. These results showed that the SPI experts involved in the evaluation stage generally did not have difficulty understanding and learning these proposed models. Neither did they have any problems understanding the workflow, i.e. the series of activities associated with each stage of the proposed models.

For the second question, the SPI expert reviewers were asked to answer the question using a scale of 1 to 5, with 5 being "too much knowledge" and 1 being "not at all." We observed that four out of five experts selected 1 or 2 on this scale for both models.

The second question was about the range of CMMI knowledge that is required to use the proposed model. In this question, the SPI expert reviewers were asked to answer the question using a scale of 1 to 5, with 5 being "too much knowledge" and 1 being "not at all." We observed that four out of five experts selected 1 or 2 on this scale for the models that were proposed for SP 1.1, SP 1.2, SP 1.3 and SP 1.5. In addition, for the model proposed for SP 1.4, two experts selected "1," two experts selected "2," and one expert selected "3" on the same scale.

Thus, it can be noted that the SPI experts directly learned how to use and adopt the proposed model. Based on the responses to the first and second questions related to "ease of learning and ease of use," it can be concluded that the proposed models were generally both clear and easy to understand, as the experts didn't require a lot of CMMI knowledge to understand them. It is important to mention here that the SPI expert reviewers believed that dividing each of the proposed models into five core stages helped them quickly understand these proposed models. This was encouraging, as it meant that we had created models that were both comprehensive and concise.

In order to evaluate the proposed models against the "stakeholder satisfaction" criteria (i.e., RQ3), the experts were asked two specific questions about each model. The first one was related to the usefulness of the model in the software industry, while the second one was related to the improvements that could be gained on quality software products by adopting the model. On each proposed workflow model, the following two questions were asked:

1. How useful would it be to the software industry to use our proposed workflow model using a scale of 1 to 5, with 5 = "very useful" and 1 = "not at all?"

2. The use of our proposed workflow model would improve the software process and lead to the production of high-quality software products. (Use the five-point scale of "strongly agree" to "strongly disagree.")

After collecting the replies to the first question, we noted that all the expert reviewers selected either "5" or "4" for the models that were proposed for SP 1.1 and SP1.2 on the five-point scale, with 5 = "very useful" and 1 = "not at all. Moreover, four out of five expert reviewers selected "5" or "4" on the same scale for the models that were proposed for SP 1.3, SP 1.4 and SP 1.5. These results show that all of the proposed models are useful to the software industry. They can also help software organizations implement SP 1.1, SP 1.2, SP 1.3, SP 1.4 and SP 1.5 of the REQM process area at CMMI Maturity Level 2 quickly.

When it came to the second question, for the proposed model for SP 1.1, two of the expert reviewers selected "strongly agree," and the other three experts selected "agree." For the model proposed for SP 1.2, four experts chose "agree," and one selected "neutral" on the same scale. For the models that were proposed for SP 1.3, SP 1.4 and SP 1.5, most (80%) of the experts had the opinion that the use of these two proposed workflow models would improve the software process as well as lead to the production of high-quality software products. These results showed that utilizing the proposed models could enhance the software process. In addition, such models will assist in producing a high-quality software product. Based on the collected responses to the first and second questions, it can be concluded that the proposed models ensure there is "user satisfaction."

We also examined the "applicability of the models to small- and medium-sized software development organizations" (i.e., RQ4), and asked the expert reviewers to rate every one of the proposed models by using our five-point scale to make a rating of this statement: "Our proposed workflow model is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations."

For the proposed model for SP 1.1, all of the experts chose "strongly agree" or "agree." For the model proposed for SP 1.2, two experts selected "strongly agree," two selected "agree," and one selected "neutral." For the proposed models for SP1.3 and SP 1.5, one expert selected "strongly agree," three selected "agree," and one selected "neutral." For the proposed model for SP1.4, three experts selected "agree," and two experts selected "neutral." Overall, the results of the expert evaluation showed that our proposed models are applicable to small and medium-sized software development organizations. This means that they can be applied to both small and medium-sized software development organizations.

Finally, according to the evaluation, we are assured that our proposed models are easy to use and learn. Moreover, the proposed models can assist small- and medium-sized software development organizations in implementing SP 1.1, SP 1.2, SP 1.3, SP 1.4, and SP 1.5 of the REQM process area, as stated by CMMI v1.3 specifications. Our models can also be applied to such organizations. The SPI experts have also asserted that our proposed models are useful, as they would help increase the depth of knowledge of SPI practitioners on the requirements management process area of CMMI Level 2.

Evaluation criteria		Evaluation results									
		SP 1.1		SP 1.2		SP 1.3		SP 1.4		SP 1.5	
		# of Experts	**Selection**	**# of Experts**	**Selection**	**# of Experts**	**Selection**	**Number of Experts**	**Selectio n**	**Number of Experts**	**Selection**
Practice satisfaction	The proposed model satisfied the goal of the specific practice, according to CMMI v1.3 specifications (strongly agree – strongly disagree)	(2) Senior Experts (1) Intermediate Expert	"Strongly agree"	(1) Senior Expert	"Strongly agree"	(1) Senior Experts (1) Intermediate Expert	"Strongly agree"	(1) Senior Expert	"Strongl y agree"	(2) Senior Experts (2) Intermediate Experts	"Agree"
		(1) Intermediate Expert (1) Junior Expert	"Agree"	(1) Senior Expert (2) Intermediate Experts (1) Junior Expert	"Agree"	(1) Senior Expert (1) Intermediate Expert	"Agree"	(1) Senior Expert (2) Intermediate Expert	"Agree"	(1) Junior Expert	"Neutral"
						(1) Junior Expert	"Neutral"	(1) Junior Expert	"Neutral"		
Ease of learning and ease of use (*This provides the answer to RQ2*)	How clear is the representation of the proposed model? (Not at all 1 - 5 Very Clear)	(1) Senior Expert (1) Intermediate Expert	"5"	(1) Senior Expert (1) Intermediate Expert (1) Junior Expert	"5"	(1) Senior Expert (1) Junior Expert	"5"	(1) Intermediate Expert	"5"	(2) Senior Expert (2) Intermediate Experts	"4"
		(1) Senior Expert (1) Intermediate Expert	"4"	(1) Senior Expert (1) Intermediate Expert	"4"	(1) Senior Expert (2) Intermediate Experts	"4"	(2) Senior Experts (1) Intermediate Expert	"4"	(1) Junior Expert	"3"
		(1) Junior Expert	"3"					(1) Junior Expert	"3"		
	How much knowledge of CMMI is required to learn how to use the proposed model? (Not at all 1 - 5 Too much knowledge)	(2) Senior Experts (1) Intermediate Expert	"1"	(1) Senior Expert (1) Intermediate Expert	"1"	(1) Senior Expert (2) Intermediate Experts	"1"	(1) Senior Expert (1)Intermediate Expert	"1"	(1) Senior Expert (1) Junior Expert	"1"
		(1) Intermediate Expert	"2"	(1) Senior Expert (1) Junior Expert	"2"	(1) Senior Expert	"2"	(1) Senior Expert (1) Intermediate Expert	"2"	(1) Senior Expert (1) Intermediate Expert	"2"
		(1) Junior Expert	"3"	(1) Intermediate Expert	"3"	(1) Junior Expert	"3"	(1) Junior Expert	"3"	(1) Intermediate Expert	"3"
User satisfaction (*This provides the answer to RQ3*)	How useful would it be to the software industry to use the proposed model? (Not at all 1 - 5 Very useful)	(1) Intermediate Expert	"5"	(1) Senior Experts (1) Intermediate Expert	"5"	(1) Senior Expert (1) Intermediate Expert	"5"	(1) Intermediate Expert	"5"	(1) Senior Expert	"5"
		(2) Senior Experts (1) Intermediate Expert (1) Junior Expert	"4"	(1) Senior Expert (1) Intermediate Expert (1) Junior Expert	"4"	(1) Senior Expert (1) Intermediate Expert	"4"	(2) Senior Expert (1) Intermediate Expert	"4"	(1) Senior Expert (2) Intermediate Expert	"4"
						(1) Junior Expert	"3"	(1) Junior Expert	"3"	(1) Intermediate Expert	"3"
	The use of the proposed model would improve the software process and lead to the production of high	(1) Senior Expert (1) Junior Expert	"Strongly agree"	(2) Senior Expert (2) Intermediate Expert	"Agree"	(1) Senior Expert (2) Intermediate Experts	"Strongly agree"	(1) Senior Expert	"Strongl y agree"	(1) Senior Expert (1) Intermediate Expert	"Strongly agree"

	quality products (strongly agree – strongly disagree)	(1) Senior Expert (2) Intermediate Experts	“Agree”	(1) Junior Expert	“Neutral”	(1) Senior Expert	“Agree”	(1) Senior Expert (2) Intermediate Expert	“Agree”	(1) Senior Expert (1) Junior Expert	“Agree”
						(1) Junior Expert	“Neutral”	(1) Junior Expert	“Neutral”	(1) Intermediate Expert	“Neutral”
Applicability of the models to small- and medium-sized software organizations (*This provides the answer to RQ4*)	The proposed model is applicable to small- and medium-sized software development organizations. (strongly agree – strongly disagree)	(1) Intermediate Expert	“Strongly agree”	(1) Senior Expert (1) Intermediate Expert	“Strongly agree”	(1) Intermediate Expert	“Strongly agree”	(2) Senior Expert (1) Intermediate Expert	“Agree”	(1) Intermediate Expert	“Strongly agree”
		(2) Senior Expert (1) Intermediate Expert (1) Junior Expert	“Agree”	(1) Senior Expert (1) Intermediate Expert	“Agree”	(2) Senior Expert (1) Intermediate Expert	“Agree”	(1) Intermediate Expert (1) Junior Expert	“Neutral”	(2) Senior Expert (1) Intermediate Expert	“Agree”
				(1) Junior Expert	“Neutral”	(1) Junior Expert	“Neutral”			(1) Junior Expert	“Neutral”

Table 13:Summary of evaluation results

5.3.2 The novelty of the proposed models

It is important to underline here that the major differences that exist between the proposed model for SP1.1 and the earlier models found in the literature suitable for understanding the requirements are presented in Table 14. Likewise, Table 15 shows the key differences between the proposed model for SP1.2 and previous models suitable for obtaining the commitment to the requirements. In addition, the main differences that exist between the proposed model for SP1.3 and the earlier models found in the literature suitable for requirements change management are highlighted in Table 16. Likewise, Table 17 presents the major differences between the proposed model for SP1.4 and previous models suitable for requirements traceability.

Table 14: The main differences that exist between the proposed model for SP 1.1 and the earlier models found in the literature suitable for understanding the requirements

	Reference			The proposed model for SP 1.1
	[48]	[47]	[27]	
A model that is staged or phased	√	√	√	√
Developing the model according to the objectives of SP 1.1 stated in the CMMI specifications.				√
Addressing the REQM process area at specific practice level				√
For small- and medium-sized organizations in particular				√
The development of the model based on specific criteria(i.e., ease of use, stakeholders' satisfaction)				√
The evaluation of the model regarding "practice satisfaction,""ease of learning and ease of use,"				√

"user satisfaction," and "applicability to small- and medium-sized software development organizations" performed				
Templates/forms/checklists presented				√
Guidelines that aim to help users go through the model provided				√
The main steps of the model are linked to corresponding templates, checklists, and forms				√

Table 15: The main differences that exist between the proposed model for SP 1.2 and the earlier models found in the literature suitable for obtaining the commitment to the requirements

Criteria for Comparing Models	Reference			The proposed model for SP 1.2
	[55]	[56]	[83]	
A model that is staged or phased	√	√	√	√
Developing the model according to the objectives of SP1.2 stated in the CMMI specifications.				√
Addressing the REQM process area at specific practice level			√	√
For small- and medium-sized organizations in particular				√
The development of the model based on specific criteria(i.e., ease of use, stakeholders' satisfaction)				√
The evaluation of the model regarding "practice satisfaction,""ease of learning and ease of use," "user satisfaction," and "applicability to small- and medium-sized software development organizations" performed				√
Templates/forms/checklists presented				√
Guidelines that aim to help users go through the model provided				√
The main steps of the model are linked to corresponding templates, checklists, and forms				√

Table 16: The main differences that exist between the proposed model for SP1.3 and the earlier models found in the literature suitable for requirements change management

Criteria for Comparing Models	Reference						The proposed model for SP 1.3
	[84]	[85]	[86]	[87]	[88]	[83]	
A model that is staged or phased	√	√	√	√	√	√	√
Developing the model according to the objectives of SP1.3 stated in the CMMI specifications.				√			√
Addressing the REQM process area at specific practice level				√		√	√
For small- and medium-sized organizations in particular				√			√
The development of the model based on specific criteria(i.e., ease of use, stakeholders' satisfaction)				√			√
The evaluation of the model regarding "practice satisfaction,""ease of learning and ease of use,""user satisfaction," and "applicability to small- and medium-sized software development organizations" performed				√			√
Templates/forms/checklists presented							√
Guidelines that aim to help users go through the model provided							√
The main steps of the model are linked to corresponding templates, checklists, and forms							√

Table 17: The main differences that exist between the proposed model for SP1.4 and the earlier models found in the literature suitable for requirements traceability

Criteria for	Reference	The

Comparing Models	[89]	[90]	[83]	proposed model for SP 1.4
A model that is staged or phased	√	√		√
Developing the model according to the objectives of SP1.4 stated in the CMMI specifications.				√
Addressing the REQM process area at specific practice level			√	√
For small- and medium-sized organizations in particular				√
The development of the model based on specific criteria(i.e., ease of use, stakeholders' satisfaction)				√
The evaluation of the model regarding "practice satisfaction,""ease of learning and ease of use,""user satisfaction," and "applicability to small- and medium-sized software development organizations" performed				√
Templates/forms/checklists presented				√
Guidelines that aim to help users go through the model provided				√
The main steps of the model are linked to corresponding templates, checklists, and forms				√

Table 18: The main differences that exist between the proposed model for SP1.5 and the earlier models found in the literature suitable for ensuring the alignment between project work and the requirements

	Reference		The proposed model for SP 1.5
	[48]	[57]	
A model that is staged or phased	√	√	√
Developing the model according to the objectives of SP 1.5 stated in the CMMI specifications.			√
Addressing the REQM process area at specific practice level		√	√
For small- and medium-sized organizations in particular			√
The development of the model based on specific			√

criteria(i.e., ease of use, stakeholders' satisfaction)			
The evaluation of the model regarding "practice satisfaction,""ease of learning and ease of use," "user satisfaction," and "applicability to small- and medium-sized software development organizations" performed			√
Templates/forms/checklists presented			√
Guidelines that aim to help users go through the model provided			√
The main steps of the model are linked to corresponding templates, checklists, and forms			√

It is necessary to mention here that the main novelties of our work need to be stressed in the following ways. First, we proposed high-level models for SP 1.1, SP 1.2, SP 1.3, SP 1.4, and SP 1.5. These proposed models fulfill the CMMI v1.3 specifications. These models were developed according to the objectives of each specific practice stated in the CMMI v1.3 specifications. Second, the requirements management process area is addressed at the specific practice level. Third, the proposed models for SP 1.1, SP 1.2, SP 1.3, SP 1.4, and SP 1.5 are designed mainly for small- and medium-sized software development organizations.

We also provided the evaluation of these proposed models regarding "practice satisfaction,""ease of learning and ease of use,""user satisfaction" and "applicability to small- and medium-sized software development organizations." It should finally be pointed out that the guidelines are another main innovation of our work that we presented. These include the templates, checklists, and forms that users can utilize to traverse our proposed models. This was performed because O'Regan [27] said, "Templates support the process and allow consistent input and output during the different parts of the process." The author further indicated that checklists can be utilized as an aid to conduct a process. In addition, Afrooz [94] said, "The template would provide the organization with a guideline to achieve the organizational objectives."Moreover, [95] gave an example that shows how a specific practice in CMMI can be implemented by presenting templates, checklists, and forms.

5.4 Remarks

The main aim of this chapter is to develop high-level models for SP 1.1, SP 1.2, SP 1.3, SP 1.4, and SP 1.5 of REQM at CMMI Maturity Level 2 specifically for small- and medium-sized software development organizations. This aim also includes the evaluation of these models using an expert panel review process. For this purpose, RQ1, RQ2, RQ3, and RQ4 were addressed.

In order to tackle RQ1, workflow models for SP 1.1, SP 1.2, SP 1.3, SP 1.4, and SP 1.5 were developed. A list of guidelines and satisfactory models were identified from the literature on each specific practice. During the literature review, many research articles and case studies in the field of SPI, including SPI models such as CMMI and software requirements engineering, were explored. We also examined and evaluated the collected data from the literature review in order to come up with the proposed models. It is important to note here that the comments and suggestions proposed by expert reviewers were utilized to improve and enhance the proposed models.

The evaluation of the proposed workflow models was conducted using an expert review panel process to tackle RQ2, RQ3, and RQ4. For RQ2, the expert reviewers were asked specific questions about the clarity of the proposed models' representation and the range of knowledge required to successfully utilize them. Regarding RQ3, two questions were clearly defined in the evaluation form in order to help us evaluate our proposed models against the "stakeholder satisfaction" criteria. The first one was related to overall usefulness of the proposed models in the software industry, while the second one was about the improvements that could be obtained in delivering high-quality software by adopting the proposed models. For RQ4, the experts were asked a particular question about the applicability of the proposed models to small- and medium-sized software development organizations.

In short, the overall evaluation results showed that the proposed models ensured practice satisfaction according to the CMMI Maturity Level 2 requirement. They also satisfied the criteria for ease of learning and ease of use; in other words, they are clear, easy to use, and easy to learn. Furthermore, they met the stakeholders' expectations and desired satisfaction level. Moreover, small- and medium-sized software development organizations can adopt the proposed models in their environments, as they are designed to be applicable to such organizations.

CHAPTER SIX: PROCESS AND PRODUCT QUALITY ASSURANCE (PPQA)

6.1 Process and Product Quality Assurance (PPQA) Overview

The successful performance of activities that are related to PPQA is an essential predictor of project success. It has been found that CMMI Level 2 as an SPI model plays a major role in improving the practices (including PPQA activities) of software projects carried out by small- and medium-sized software companies. PPQA is one of the fundamental process areas of CMMI Level 2. As stated in the CMMI v1.3 specifications [65], the purpose of PPQA is to "provide management and staff with objective insight into processes and related work products." If the PPQA process area is applied successfully, it will result in the delivery of high-quality software products. Chrissis et al. [17] indicated that PPQA gives staff, personnel, and managers valuable insights while conducting PPQA activities. Khraiwesh [96] highlighted that the PPQA PA aims to ensure the quality of the software product by monitoring the process, procedures, and techniques that are used to conduct the software project. This is also a process that can be used to confirm and ensure whether products or services will satisfy user expectations. Jarvis et al. [25] indicated that PPQA is an essential process for both the software developer and software buyer, as it is the procedure that determines whether the techniques and methods applied have been correctly integrated and performed. O'Regan [27] indicated that PPQA offers visibility to the software project management during the software development process, as well as related work products developed by the project.

Table 19 Process and Product Quality Assurance (PPQA) process area

CMMI Specific Goal	CMMI practice number	CMMI practice description
SG 1	Objectively evaluate processes and work products	
	SP 1.1	Objectively evaluate processes
	SP 1.2	Objectively evaluate work products
SG 2	Provide objective insight	
	SP 2.1	Communicate and resolve noncompliance issues
	SP 2.2	Establish records

As described in Table 19, the PPQA process area involves four specific practices, and achieving them greatly facilitates the delivery of high-quality software products. This is clearly because, throughout the software development life cycle, these practices offer objective insights to the software project managers (PMs). In addition, they provide appropriate feedback on both the software processes and related project work products.

Different empirical studies have been carried out to address the challenges that are faced by small- to medium-sized software development organizations in implementing CMMI Level 2 as an SPI model. These studies help software practitioners in such organizations to give more attention to the high perceived value of different practices of CMMI Level 2 process areas. This information enables such organizations to develop their own finer-grained CMMI Level 2 framework, which will definitely help small- to medium-sized software development organizations to implement SPI initiatives in a better way.

- Wilkie et al. [62] showed that SP 1.2 was the most important of the various PPQA specific practices. The second, third, and fourth most important were SP 1.1, SP 2.1, and SP 2.2, respectively. Their findings also emphasized that most of the companies involved in conducting the research showed poor performance in both monitoring and controlling process usage. Consequently, such companies have very little ability to identify any noncompliance issues.
- Niazi et al. [26] and Niazi et al. [34] highlighted that less than 30% of developers and managers reported the value of all specific practices of the PPQA process area as high. However, more than half of them cited the relative perceived value of these specific practices as medium.
- Lester et al. [63] showed that medium-sized companies covered all specific practices of the PPQA process area better than small ones. Their findings also highlighted that when it came to average values, SP 1.2 had the highest value of the practices used for the PPQA process area by both small- and medium-sized software development organizations. SP 1.1, SP 2.1, and SP 2.2 were the second, third, and fourth highest, respectively. Their findings also revealed that larger companies had enough resources to invest in order to perform the PPQA

PA. However, this process area is regarded as overhead for small- and medium-sized software development organizations due to their limited resources.

Hence, more attention should be paid to all specific practices of the PPQA process area. In other words, there is a need to give careful consideration to these practices and concentrate on them more. It is evident that a project's success or failure is strongly tied to objectively evaluating the processes and work products for it and offering objective insights to project managers. However, not enough research studies have been carried out to build the effective implementation of all specific practices of the PPQA process area at CMMI Level 2. Thus, in this research work, we have proposed a workflow model for each specific practice, that is, SP 1.1, SP 1.2, SP 2.1, and SP 2.2 in the PPQA process area.

6.2 Recommendations for implementing the specific practices of the PPQA

This section discusses our findings from the literature, including both guidelines and recommendations for evaluating and assessing the performance and quality of the processes, services, and the project's work products. In addition, suitable process maps or workflow models that would assist in providing an appropriate assessment of the software development processes and managing the process throughout its lifecycle are presented.

6.2.1 Recommendations for implementing SP 1.1 and SP 1.2

Different research studies have been undertaken to objectively evaluate processes and work products:

- Persse [64] recommended the following activities to support and implement both **SP 1.1 and SP 1.2**: (1) selecting key processes and products based on established selection criteria, which will greatly help in the auditing of those processes and work products having a serious impact on the success of the software project; (2) establishing a well-described audit procedure that can be used by the PPQA auditor each time he or she tackles the audit procedure; and (3) establishing a common PPQA plan that can be used as a reference because it contains important information entries and item, such as who is required to participate and how audits will be conducted.
- Chrissis et al. [17] provided examples of evaluation criteria that should be established and maintained for **SP 1.1 and SP 1.2**, that is, which software project items should be evaluated when the software process/work product is examined, how a software process/work product is assessed, and who is accountable for the evaluation process/work product. In addition, the authors further highlighted that selection criteria should be declared and properly stated before performing the audit process. Moreover, noncompliance issues and

lessons that are learned should be well-identified to improve the process/work product.

- Moorthy [81] wrote a book titled *CMMI Implementation Guide: A Practitioner's Perspective*, which can be considered as a reference book for software development organizations that have an interest in implementing CMMI in their environments. In this book, the author proposed the process review report (including corrective actions and audit checklists) as a typical artifact that can be used to satisfy **SP 1.1**. For **SP 1.2**, the work product review report can be used as a CMMI artifact.

6.2.2 Recommendations for implementing SP 2.1

Various research studies have been performed to better communicate and resolve noncompliance issues:

- Persse [64]advocated the following activities for the implementation of **SP 2.1**: (1) keeping the goal in mind in order to judge the importance of the identified noncompliance issue against the objectives and goals of the project, (2) communicating the value of compliance to team members to assist in encouraging success and supporting the project's further progress, and (3) following up with support to ensure that identified noncompliance issues are resolved.

- Chrissis et al. [17] highlighted the importance of fixing or working out each noncompliance issue by involving the right team members. If the noncompliance issue cannot be fixed, it should be well-documented and escalated to the proper project management level to control the issue and track it to its resolution. In addition, the audit reports, including the evaluation results, should periodically be published for the relevant stakeholders.
- Moorthy [81] proposed the noncompliance closure report as a typical CMMI artifact that could be used by organizations to satisfy **SP 2.1**.

6.2.3 Recommendations for implementing SP 2.2

Various studies have been done to effectively establish PPQA records:

- Persse [64] highlighted some activities that can help the PPQA auditor/reviewer to satisfy **SP 2.2**. For example, the PPQA records should indicate the status of the audit process and reflect its results, which must be well-reported to project management, team members, and organizational management. It is important to keep in mind that the PPQA records offer objective insights to project management on the process/work product that is to be audited. They also show how the team members have performed and how they have worked within organizational standards so that project management will be able to generate appropriate and relevant appropriate comments. In addition, reporting and sharing the PPQA records among the team members will result in increasing the number of objective insights, not only for project management, but also for those who are responsible for executing the project's activities. Moreover, the presentation or establishment of records is another major factor that impacts the audit process. Thus, the PPQA records should be presented in an appropriate manner that captures all of the necessary data. This could be achieved by using graphs, tables, scorecards, and snapshots in time.

- Chrissis et al. [17] indicated that PPQA activities have to be recorded in an adequate manner in order to recognize the status of the audit process and its results. In addition, the history of PPQA activities should be updated in a timely manner.
- Moorthy [81] considered the status of corrective actions, the audit plan, and the noncompliance closure report to be the main CMMI artifacts for **SP 2.2**.

6.3 Suitable models for the specific practices of the PPQA PA

In addition to the above studies, some models and process maps to better provide feedback on both the processes and related work products have also been proposed. The following models and process maps that address how the process's quality is evaluated (including the auditing and reviewing process) were identified from the literature. It is believed that these process maps can be adjusted and modified to implement the specific practices of the PPQA process area.

- **The four-stage model for auditing a process/work product (Audit-Guide [97])**

In [97], it is indicated that the audit process involves four core stages: planning, execution, reporting, and follow-up (see Figure 16). In this model, the main aim of the **planning** stage is to select the process or work product that is to receive an in-depth assessment and audit. It is necessary to specify the evaluation criteria for this selection so that the evaluation results can be used to reach a conclusion about the effectiveness and success of the process/work product. After the selection is made, pre-study activity takes place to find out the necessary background knowledge about the selected item. The audit process is conducted in the **execution** stage, which involves several major activities, such as collecting data and analyzing evidence. The audit evidence can be obtained in several ways, such as direct observation of events, conducting interviews or questionnaires, and reviewing the written documents, such as reports, plans, and records.

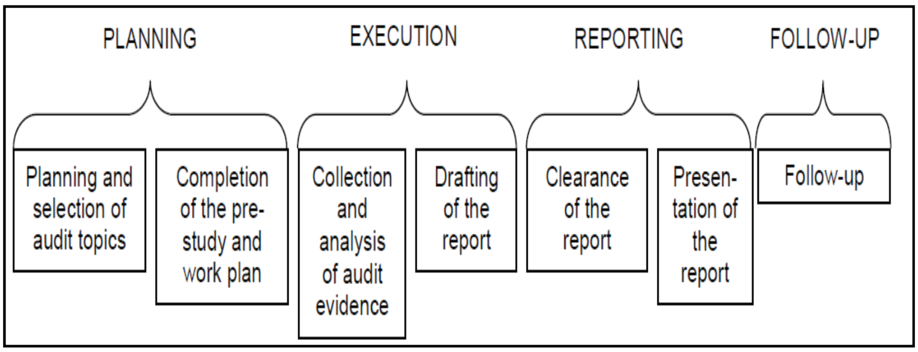

Figure 16 Stages involved in auditing and reviewing a process/work product [97]

It is worth mentioning here that the execution stage works toward developing the findings and deriving the conclusions. Consequently, to derive the audit findings and identify their impact on the project's success, the answer to the question of what should be done needs to be compared with what has been done. The final activity is providing a draft report about what has been performed and obtained throughout the execution stage. In the **reporting** stage, the manager is involved in reviewing the quality of the draft report. After that, the final version of the report is produced. In other words, the main concern at this stage is to conduct the clearance examination of the generated report via a review process, meetings, and quality control.

The final version of the report is submitted to both concerned stakeholders and users. Finally, **follow-up** is a core stage in the audit/review process, as it reports on the audit's impact. It also identifies the progress that has been made in implementing the corrective actions and audit recommendations.

- **Process audit workflow model (Seattle Area Software Quality Assurance Group [SASQAG] [98])**

The model starts with the establishment of an audit plan, which should include both the requirements and checklists. The success of the audit process is directly proportional to the preparation stage. The preparation assists greatly in indicating which audit items, such as the process areas/work products, have had a significant impact on the project's success.

It is also useful to concentrate on the specific processes/work products. Therefore, the audit/review process is conducted, and the PPQA auditor should both report on the findings and document them. The project manager is involved in reviewing and examining the findings. If noncompliance issues are identified, then corrective actions need to be developed to work out these issues.

The corrective action stage should include the following activities: determining the action, establishing the corrective action plan, controlling it to completion, and analyzing the impact of the corrective action. The follow-up stage helps the auditor to keep track of the audit actions. Immediately after an audit action has ended, this stage assists the auditor in bringing its status up to date so that it can be closed. The PPQA auditor should check the actions and close any noncompliance issues.

- **Process activities and flowchart for PPQA (Florina-Cristina et al. [99])**

Florina-Cristina et al. [99] proposed a flowchart/process map for the whole PPQA, which shows the process activities that need to be performed to implement the PPQA process area activities. The flowchart comprises seven core activities: (1) establishing PPQA selection criteria, instructions, and planning; (2) detailed planning of PPQA; (3) objectively evaluating the processes and documenting the results; (4) objectively evaluating work products and documenting the results; (5) communicating the quality status; (6) tracking, managing, and resolving quality problems and escalating them if necessary; and (7) establishing, maintaining, and deploying PPQA records.

The authors highlighted the importance of preparing the following documents: a set of questions that will work as the evaluation criteria, an audit schedule, and audit instructions. The audit/review process should then be conducted, based on the specified criteria and defined instructions.

The authors have further stressed the major role of documenting and analyzing noncompliance issues. In addition, templates and checklists should be used in an appropriate manner as tools to track and schedule noncompliance issues and corrective actions. This needs to be performed to indicate the due date for implementing the corrective actions and collecting improvements. The proposed process map for the PPQA process area ensures that not only will the reporting of noncompliance issues take place, but also that the evaluation process will be completed according to the specified evaluation criteria.

- **Audit process workflow model (Pankaj [100])**

Pankaj [100] introduced a review/audit process workflow model with four main stages: Planning, Preparation and overview, Group review meeting, and Reworking and follow-up. In this model, the planning stage involves selection of the reviewers/inspectors who will conduct the auditing process. It also establishes the schedule for the review process. This stage should highlight the selected reviews and schedule for the auditing process. In addition, a full package that includes all of the necessary documents is distributed to the reviewers. In the second stage, a short seminar on the selected process/work product for review could be provided, if necessary, to clarify some of the issues that are difficult to understand. Throughout this stage, any specific issues about the process/work product are clearly illustrated and pointed out.

After that, the final copy of the package is submitted to the reviewers/inspectors. The self-review/self-preparation process is then performed by making notes whenever a defect is observed. Before conducting the third stage, it is necessary to check whether the reviewers/inspectors performed the self-review process.

The aim of the third stage is to discuss the findings, analyze the audit evidence, and produce the list of defects. Reviewers/inspectors are required to contribute to the meeting. The author stressed that the main focus of this stage is to identify the problems in the software project but not to come up with a solution to resolve them. The main output of this stage includes the defect log, derived conclusions, recommendations, and completed review/audit report.

In the last stage, that is, reworking and follow-up, the author of the project work should conduct the rework stage, based on the recommendations that were produced in the third stage. In addition, he or she needs to correct all of the reported defects and then review the corrections made. This stage should also generate the summary report and lessons learned.

- **Process audit workflow model (Westfall [101])**

Westfall [101] proposed a process map for the audit process that includes six primary stages: (1) Initiation, (2) Planning, (3) Preparation, (4) Execution, (5) Reporting, and (6) Corrective actions and follow-up. In this proposed audit process map, the client, such as an external customer or internal manager, performs the formal initiation of this process by requesting an audit. It is important to highlight here that the frequency of conducting the audit process mainly depends on the objectives of the software development organization and the regulatory or contractual audit requirements.

After that, the auditor management, auditee management, and lead auditor need to cooperate to establish the audit plan, which should be documented and communicated to all audit team members. One of the main objectives of the planning stage is to select a suitable strategy to execute the audit process.

Following that, the auditors are requested to prepare for the audit/review process, which includes studying the process/work product to be audited, as well as gathering and exploring the related information/documents. After the preparation stage is completed, the real execution of the audit process is carried out in the execution stage. The objective evidence is collected at this stage by implementing the audit plan and utilizing the audit tools and audit checklists. The auditors need to evaluate the equality records, examining the related documents, looking for patterns by studying data or software matrices, conducting interviews, and assessing the physical properties. The lead auditor is responsible for reporting the audit findings in the audit report. This report should include detailed findings, nonconformance issues, and opportunities for improving the process/work product. Best practices, observations, and conclusions should also be reported.

Finally, the corrective actions and follow-up stage is conducted by the auditee management and the lead auditor. They are responsible for establishing and planning corrective actions for any nonconformance that is identified during the audit process and also for determining whether this action is effective and successfully generates the desired results.

Once corrective action implementation is completed, the lead auditor takes care of verifying this implementation. This is done to ensure that the nonconformance issue is fixed correctly by testing the implementation of the proposed corrective action and carefully reviewing the associated documentation. In general, this verification can be performed by conducting follow-up audit activity and reviewing the reexamined documents. The follow-up audit activity will also assist in tracking the corrective action until it is closed.

- **Process review/audit workflow model (Khurana [102])**

Khurana [102] has proposed a generic workflow model that assists in the auditing/reviewing of a process/work product (see Figure 17).. This model starts with the preparation that is required for developing a software product review process. This preparation phase can involve various activities, such as defining the review/audit criteria and selecting the process/documents to audit. Copies of the documents produced during the preparation phase are distributed among the software reviewers.

Before conducting the official review/audit meeting, the reviewers may need to make individual preparations by examining the documents and writing down their comments. After that, the review/audit meeting phase is used to perform the evaluation and derive both the conclusions and recommendations. This phase also involves reporting the results and making the required decisions for controlling and managing any software defects. If another review meeting is needed, documents are returned to the preparation phase for additional understanding, observation, and evaluation.

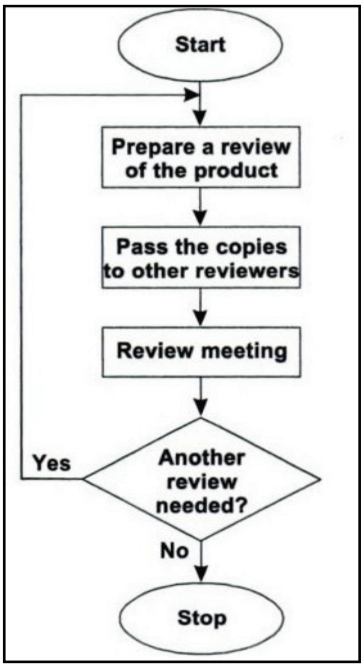

Figure 17 review/audit process workflow model (Khurana [102])

- **Sample process map for corrective action process [103]**

In [103], it is indicated that a software project team is required to tackle software quality assurance (SQA) by establishing a corrective action (CA) process. This process is most probably reported in the project plan or SQA plan. It may also be documented in the organizational quality management plan. As indicated in the sample process map for the corrective action process proposed by [53], SQA auditors/reviewers need to indicate the root cause of any nonconformance issue, as well as its impact. The corrective action should be proposed by the project team to manage and deal with a nonconformance issue.

Each suggested corrective action is reviewed and audited by the SQA auditors/reviewers to ensure that it addresses the related nonconformance issue. If corrective action appropriately tackles the issue, then effectiveness measures will be identified to indicate whether the suggested corrective action can effectively resolve the issue and produce the intended results. After establishing the proposed corrective action, SQA auditors/reviewers need to evaluate the implementation of the corrective action in terms of its effectiveness. It is necessary to highlight that if the applied and executed corrective action is not effective, SQA auditors/reviewers need to go back to the stage of determining an appropriate corrective action for the purposes of additional understanding and analysis.

- **PPQA process map (O'Regan [27])**

O'Regan [27] proposed a sample process map for implementing the PPQA process area at CMMI Level 2. The proposed process map indicates that the PPQA process area should perform the following core activities: planning, conducting an auditing process, producing audit reports, examining the audit actions, and updating the audit actions to completion. O'Regan further indicated that the PPQA process area relates to the implementation of practices that target the planning, scheduling, and conducting of audits. In addition, the author has provided templates for audit reports and audit schedules that can help to ensure that activities associated with the PPQA process areas have been completed.

6.4 The proposed models for the specific practices of the PPQA process area

This section discusses the development of the proposed models for the specific practices of the PPQA process area. Our models are proposed based mainly on our findings from literature presented in section 5.2. Each model is divided into core stages. Certain activities associated with each stage are clearly indicated.

6.4.1 Description of the models proposed for SP 1.1 and SP 1.2

The models for SP 1.1 and SP 1.2 are divided into four essential stages: "Plan," "Prepare," "Audit," and "Report." Particular activities associated with each of these stages have been clearly described (see Figure 18 and Figure 19).

The first stage is called the "Plan." This stage is included in both models, as has been noted in the following audit/review workflow models: SASQAG [98] and Pankaj [100]. Additionally, the "plan" is explicitly indicated as the main stage of the four-stage model for auditing and reviewing a process/work product [97]. Furthermore, O'Regan [27] introduced a sample process map for implementing the PPQA process area. In this sample, the "plan audit" is included as one of the core activities that assist in the implementation of PPQA. Furthermore, Persse [64] stressed the importance of establishing a PPQA plan and considered it to be one of the main activities that can be used to support and implement both SP 1.1 and SP 1.2.

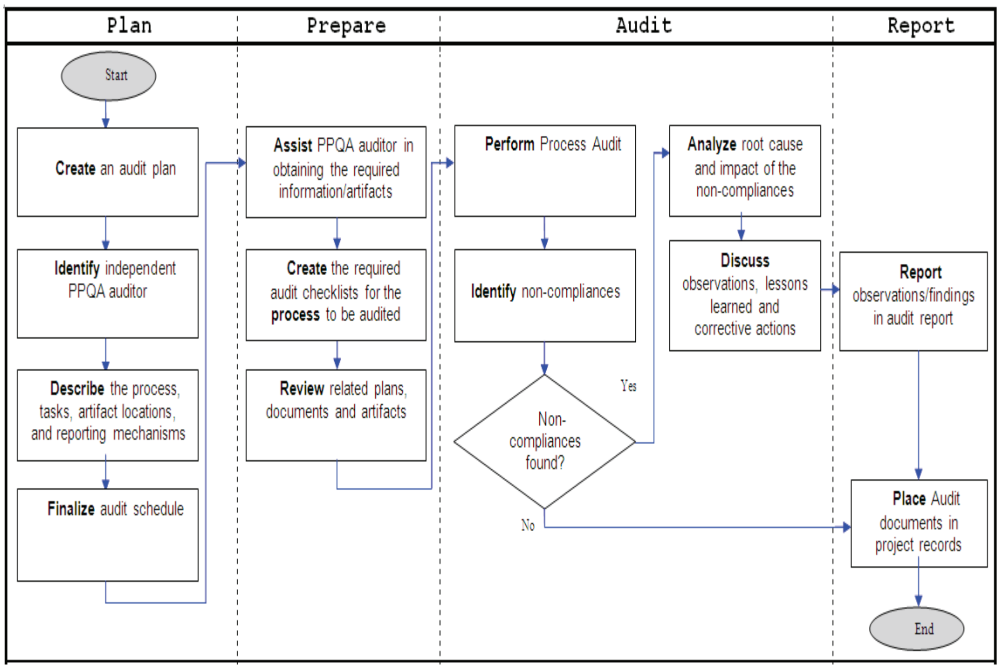

Figure 18: The proposed model for SP 1.1– "Objectively evaluate processes"

This initial step includes the detailed planning of activities, such as ensuring the availability of an organization's specific quality plan, quality policies, and process quality assurance plans, based on the organizational quality plan. This stage involves establishing quality procedures and creating quality plans that include a description of the process to be followed, as well as specifying the selection criteria to be used to determine the process/work product for the audit.

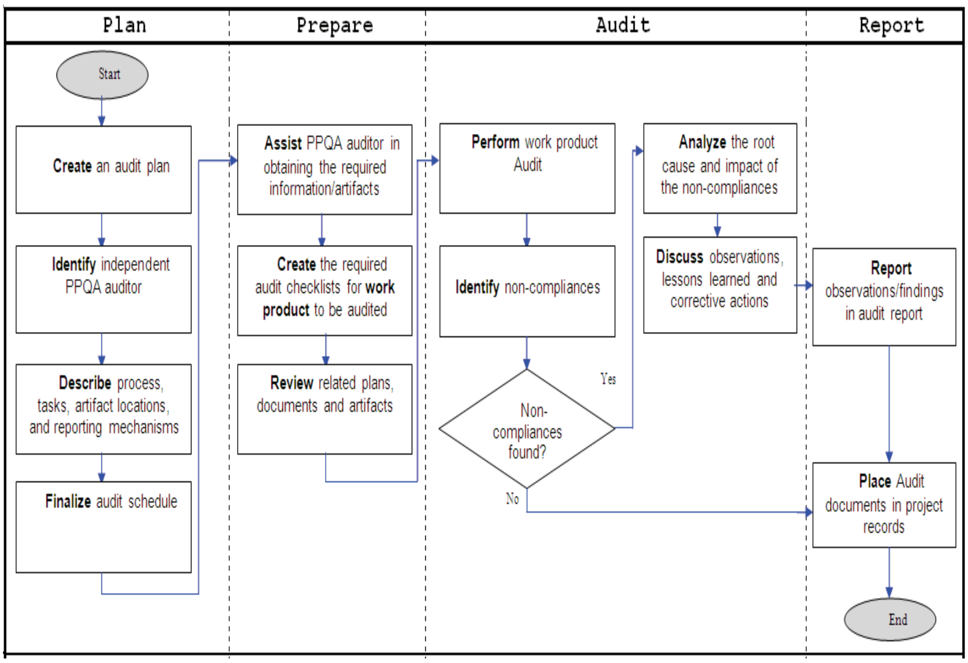

Figure 19: The proposed model for SP 1.2– "Objectively evaluate work products"

It is worth mentioning that Persse [64] pointed out the significance of establishing selection criteria, which has a serious impact on the development of a successful software project when reviewing the process/work product. Table 20 provides some of the criteria that can be used to select the key process/work product to be audited.

The created audit plan should clearly describe which process/work product is to be audited, for what purpose, at what frequency, and by whom. It should also primarily describe the procedure that is to be followed, the tasks, artifact locations, and reporting mechanisms. The project manager (PM) can assist the PPQA auditor when creating the audit/review plan for the project.

Table 20 Criteria that can be used to select a key process/work product to be audited.

No.	Question
1	Does this process/work product have a serious impact on the success of the software project?
2	Will the auditing of this process/work product help the company to achieve its mission and vision?
3	Will the auditing of this process/work product result in significant benefits?
4	Can the auditing of this process/work product reflect the needs of the company?
5	Is the data related to this process/work product collectable?
6	Is the data related to this process/work product available?
7	Are necessary recourses available to support the auditing of this process/work product?
8	Will the auditing of this process/work product be completed within predefined time frames?

It is necessary to note that the PPQA auditor needs to be knowledgeable about how the audit/review process can be conducted to objectively evaluate the selected process/work product. Accordingly, he or she should be well-trained in the organizational quality process. Table 21 presents the proposed criteria that can assist in selecting an auditor/reviewer to carry out auditing/reviewing activities. Finally, as part of the planning phase, the project PPQA auditor will finalize the audit plan with the respective PM.

The second stage is “Prepare.” This stage is incorporated into both models, as it has been highlighted in the following references cited in the literature: Audit-Guide [97], SASQAG [98], and Pankaj [100]. Furthermore, Khurana [102] and Chemuturi

[104] presented a workflow model for the audit/review process. In both models, preparation is the main stage before progressing to the actual auditing stage.

Table 21 Criteria to select the auditor/reviewer

No.	Question
1	Does this person have good documentation skills?
2	Does this person have good verbal skills?
3	Dose this person have enough knowledge to contribute to the process/work product?
4	Does this person have the ability to assess the impact of non-compliances?
5	Does this person have the appropriate training to conduct auditing activities?
6	Does this person have strong enough power to bring both problems and non-compliances to light?

The "Prepare" stage helps the PPQA auditor to be better prepared by conducting key activities, such as preparing the checklists for the process/work product to be audited and reviewing the associated documents. The auditor/reviewer needs to perform a self-preparation process by reviewing both artifacts and reports that are related to the selected process/work product. Moorthy [81] considered the audit checklists as being typical CMMI artifacts for both SP1.1 and SP1.2. In addition, O'Regan [27] emphasized the importance of the PPQA audit checklists in guiding the auditor/reviewer while conducting the auditing process. Therefore, before going any further with this stage, it should be ascertained that the audit checklists that are related to the selected process/work product are well-prepared.

The third stage is "Audit." This is the actual audit of the performed process, where the auditor identifies noncompliance issues, improvements, and best practices. The selected process/work product for auditing should be objectively evaluated by using the defined criteria to ensure that the process is implemented according to defined standards. For example, the auditor/reviewer needs to check whether the delivery goes against an approved Statement of Work (SOW) and whether all the defects revealed by testing have been closed, according to the defined thresholds.

We have proposed two audit checklists: one for process and the other for work product (see Appendix H and Appendix I). They can be used to guide the auditors while they perform the audit/review process. We have adopted the idea of a Standard CMMI Appraisal Method for Process Improvement (SCAMPI), particularly Class A appraisal. It is worth mentioning that SCAMPI is a well-known SEI process evaluation method for CMMI (Dern [106]).

SCAMPI has been developed to offer a benchmark-quality rating that is relative to the CMMI models (SCAMPI v1.3 document [105]). It is viewed as an assessment method that can be used to obtain the CMMI certification for a software development organization. This method helps to pinpoint both the strongest and the weakest process followed by an organization, and it determines the organization's related maturity level (Dern [106]; Koirala and Sheikh [108]). In other words, it examines how software development organizations implement the process areas that are defined in CMMI specifications, based on objective evidence. It also releases information about development risks. As stated in the SCAMPI v1.3 document

[105], the SCAMPI method comprises three core stages: planning and preparing for an appraisal, conducting the appraisal, and reporting the results. SCAMPI supports three audit/assessment classes, namely Classes A, B, and C. Dern [106] pointed out that Class C is regarded as an informal SCAMPI, as it is shorter than Class A and Class B. In other words, it can be used as a quick evaluation method, as less objective evidence is required in this appraisal class (Kenett et al. [107]; Koirala and Sheikh [108]).

Consequently, it can be deployed in several contexts (such as rapid analysis for the process/work product and checking the implementation of the processes). The scale that can be used for Class C appraisal has three levels: high, medium, and low. Dern [106] indicated that Class B is viewed as unofficial SCAMPI, as it is not a complete evaluation/assessment process. However, this class helps to roughly rate the level of confidence and accuracy of CMMI audit acquiescence.

An experienced or trained person can execute the appraisal process (Koirala and Sheikh [108]). Red, yellow, and green are the three points that form the SCAMPI Class B scale. Generally, this class can be utilized in preparation for Class A appraisal. As highlighted in [106]–[108], Class A appraisal corresponds to official SCAMPI, and it is the only class that can provide the rating. The process in this class is conducted by the reviewer/auditor, who is SEI certified and authorized to perform the appraisal/evaluation process. A large amount of objective evidence is required to conduct Class A. The following points compose the Class A scale: Fully Implemented (FI), Largely Implemented (LI), Partially Implemented (PI), Not Implemented (NI) and Not Yet Implemented (NY).

In this research work, each item in both audit checklists should be evaluated against the criteria of Class A appraisal, namely FI, LI, PI, NI, and NY. The criteria of Class A appraisal were utilized because they were used in the references that were cited in the literature (Sharma et al. [109]; Sharma et al. [110]; Satrio et al [111]).

It is important to highlight here that the primary focus of the “Audit” stage is to identify areas of noncompliance and improvement in the software project. “Identifying the noncompliance” activity has been defined in this stage of our models because it is explicitly indicated in the CMMI v1.3 specifications [65] and Chrissis et al. [17]. Additionally, such activity is noted in the workflow models and process maps that have been proposed for the audit/review process (O’Regan [27]; Audit-Guide [97]; SASQAG [98]; Florina-Cristina et al. [99]; Pankaj [100]; Chemuturi [104]).

In cases where noncompliance is not found, the auditor should place all the audit documents in the project records and then end the specific practice. However, if any noncompliance is identified, the auditor needs to analyze the root cause of it, as well as its impact. Discussions about the observations/findings, followed by the corrective actions that are used, should resolve any noncompliance issues. Moorthy [81] viewed the corrective actions as one of the major CMMI artifacts needed for implementation of both SP1.1 and SP 1.2. This is why they have been included as one of the major activities in this stage. It is vital to emphasize that objectivity must be maintained during the auditing/reviewing process and that any noncompliance that is detected is justified with sufficient details so that it can be closed easily. The auditor also needs to check during this stage whether the teams are implementing best practices in their project and should encourage the implementation of such practices.

"Report" is the last stage of the proposed models. The observations and findings are recorded into the audit report once they have been finalized. After that, all audit documents are placed in the project's records and the specific practice is closed. The PM might need to be involved in reviewing the audit report and then approving it. The final version of the audit report might need to be distributed to concerned stakeholders and users. This stage is defined in the models because it was noted in the references cited in the literature (O'Regan [27]; Audit-Guide [97]; SASQAG [98]; Pankaj [100]). Furthermore, the noncompliance reports are underlined as one of the basic work products for SP 1.1 and SP 1.2 in both CMMI v1.3 specifications [65] and Chrissis et al [17].

6.4.2 Description of the model proposed for SP 2.1

In this section, we have implemented the CMMI Level 2 specific practice SP 2.1 "communicate and resolve noncompliance issues" in the PPQA PA. The proposed model effectively assists in both fixing noncompliance issues and ensuring the issues that are highlighted in the audit report are addressed in a timely manner. The proposed model has four essential stages: "Resolve," "Escalate," "Follow-up," and "Find-Out." Certain activities associated with each of these stages have been clearly pointed out (see Figure 20).

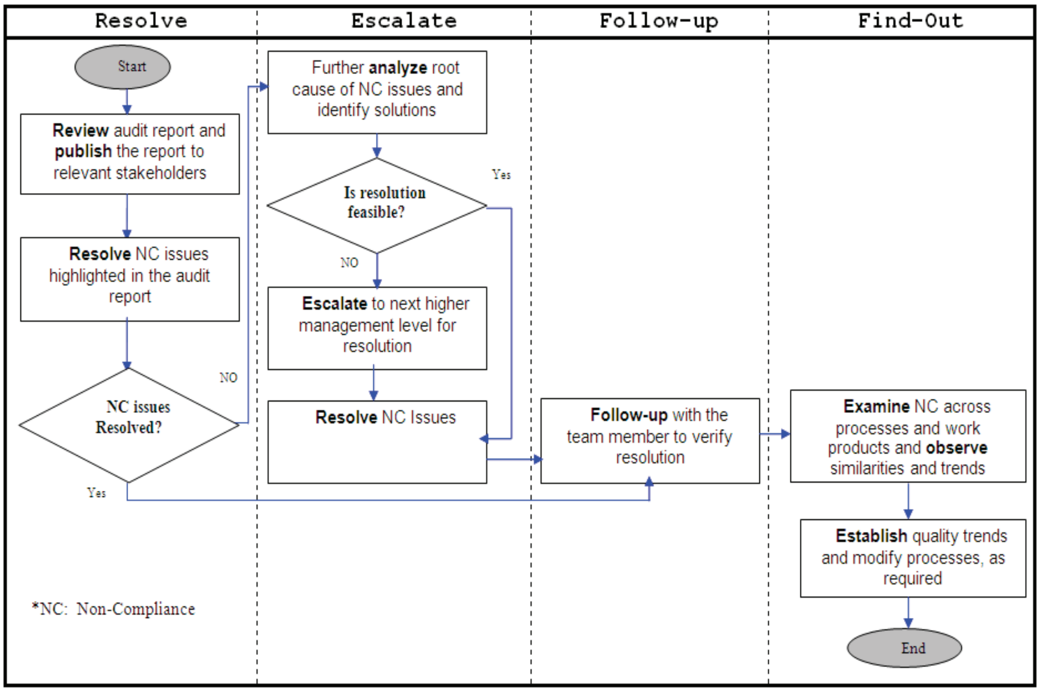

Figure 20: The proposed model for SP 2.1 –"Communicate and resolve non-compliance issues

The first stage is "**Resolve**." This stage has been defined in our model because resolving noncompliance issues is explicitly indicated in both the CMMI v1.3 specifications [65] and Chrissis et al [17] as one of the main sub practices to be carried out to implement SP 2.1. It is also seen in the explanation of the process maps that were proposed for the audit/review process by O'Regan [27], Audit-Guide [97], SASQAG [98], Florina-Cristina et al. [99], Pankaj [100], and Khurana [102]. Our proposed model starts with the activities that are needed to resolve the noncompliance issues that were identified in the evaluation of the process/work product. These issues were highlighted in the audit report that was produced in SP 1.1 and SP 1.2. Thus, the audit/review report is regarded as a primary input, and it is fully utilized during this stage. The first activity in the "resolve" stage is to review the audit report and then release it to the stakeholders for further action, if needed. Following that, the auditor starts resolving any noncompliance issues that are based on the recommendations and corrective actions provided in the audit report. If all of the noncompliance issues pointed out in the audit report are resolved and addressed, then the follow-up stage will take place. Otherwise, in cases where there is no feasible resolution to noncompliance issues, the escalation path will be taken.

The second stage is "**Escalate**." This element is included in the proposed model, as it was clearly noted in the process maps introduced by O'Regan [27] and Florina-Cristina et al. [99]. Furthermore, it is considered to be a major subpractice by both the CMMI v1.3 specifications [65] and Chrissis et al [17] for SP 2.1. When a noncompliance issue cannot be resolved, further understanding and analysis are required to determine the main source of the issue.

The solutions and necessary steps for resolution also need to be identified, and then, if the issue still cannot be resolved, it is escalated and reported to the appropriate level of authority, as defined in the project management plan. Once a resolution has been obtained, we move on to the third stage. It is worth mentioning here that the decision symbol presented in this stage is included to indicate that escalating noncompliance issues to the next management level for resolution should not occur very often. O'Regan [27] and O'Regan [112] confirmed this point.

The third stage is "Follow-up." This stage is included because it is clearly stated, in detail, as one of the primary stages in the following proposed workflow models for PPQA: O'Regan [27], Audit-Guide [97], SASQAG [98], and Pankaj [100]. In addition, CMMI v1.3 specifications [65] and Chrissis et al [17] regarded follow-up of noncompliance issues to resolution as one of the primary subpractices that needs to be performed to implement SP 2.1. Moreover, Persse [64], O'Regan [27], and O'Regan [112] stressed the significant role that the follow-up activity plays in monitoring the closure of audit actions. At this stage, regular meetings are conducted between the PPQA auditor and project team members to verify that the corrective actions that are taken to resolve noncompliance issues have been properly carried out, as well as to ensure smooth closure of these issues.

It is worth mentioning that follow-up should be conducted at an agreed/predefined time interval to give the team a reasonable amount of time to close noncompliance issues. In other words, a comfortable amount of time should be given to the team to follow correction activities through an appropriate closure. In very rare cases, an escalation mechanism will be used to act on noncompliance issues that are still not closed properly, even during the follow-up stage. When all the corrective actions are carried out and all noncompliance issues resolved, the auditor/reviewer will need to perform a final check by signifying the closure of these issues before proceeding to certify the process/work product.

"Find-Out" is the last stage in the proposed model. This stage is incorporated in the model as it is explicitly stated in the following references cited in literature: CMMI v1.3 specifications [65], Chrissis et al [17], O'Regan [27], and O'Regan [112]. In this stage, the auditor needs to perform an in-depth analysis of all the identified and closed noncompliance issues to identify any significant patterns or trends across the current project.

Observing any similarities and trends will strongly facilitate the process of addressing specific issues with processes/work products/resources to improve the software project and organizational processes. This also ensures that the instances of noncompliance will not be repeated when they are identified in a project or in other projects or business units of the organization.

For further investigation, the auditors might need to compare the analysis results obtained with other projects, as well as the expectations they may have as a result of previous experiences. This will help to tackle process improvements at the organizational level. It is vitally important for the auditor to share the results with relevant stakeholders.

6.4.3 Detailed description of the model proposed for SP 2.2

In this section, CMMI Level 2 specific practice SP 2.2 "Establish Records" has been implemented. The proposed model for this specific practice is based on our findings from the extensive literature review presented in Section 4. This model assists in the establishment of records of PPQA activities at both the project level and the organizational level. In other words, the proposed model not only provides management at the project level, but also carries out process improvement initiatives at the organization level. Our proposed model is based on four essential elements: "Record," "Revise," "Share," and "Improve" (see Figure **21**). In addition, the related activities for each of these stages have been highlighted.

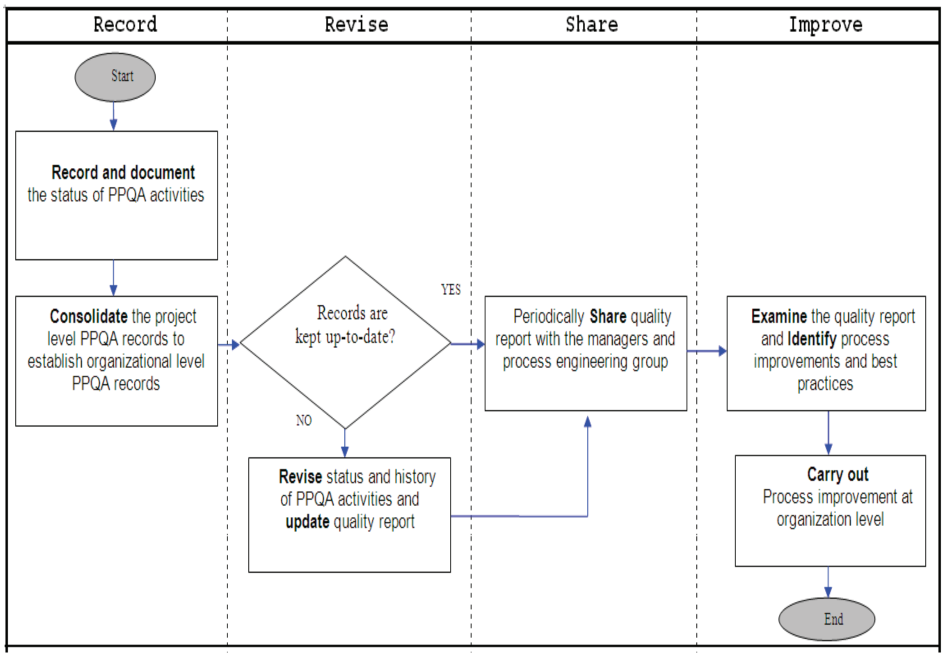

Figure 21: The proposed model for SP 2.2 –"Establish records"

"Record" is the first stage in the proposed model. The model starts with detailed records of PPQA activities at both the project and organizational levels. This stage is incorporated in our proposed model because O'Regan [27], CMMI v1.3 specifications[65] , Chrissis et al. [17], Persse [64], Chemuturi [104], and O'Regan [112] have all clearly emphasized the significant role that detailed PPQA records play in offering objective insight to software project managers about the processes, as well as related work products. Furthermore, PPQA records should be addressed at both organizational and project levels because Persse [64], Chemuturi [104], and O'Regan [112] have highlighted the importance of consolidating project-level PPQA activities to create organizational-level PPQA records.

It is worth mentioning here that SP 1.1 and SP 1.2 (the first and the second specific practices in the PPQA PA) are conducted to objectively evaluate both the selected process and work product. The observations, along with the corrective and preventive actions obtained from conducting these two specific practices, are recorded in an audit report and shared with stakeholders. It should also be noted that SP 2.1 is conducted to track the observations and close them through regular follow-ups and an escalation mechanism with the project stakeholders.

The record stage is used to ensure that the observation trends, previously obtained from SP 1.1, SP 1.2 and SP 2.1, are documented at project level. It also makes certain that the status of the records of various PPQA activities are recorded and documented at project level. At this stage, the PPQA auditor/reviewer also needs to utilize the PPQA records recorded at project level from the individual project that has been developed to create organizational-level PPQA records (also known as independent PPQA records). In other words, data from the individual project should be collected and then consolidated into a detailed organizational-level PPQA report. It is necessary to highlight here that PPQA records should present all of the important elements, such as audit reports, status report of corrective actions, and evaluation logs from individual projects.

The second stage is “Revise.” This stage is added because it is indicated as one of the primary sub practices that are proposed by CMMI v1.3 specifications [65] and

Chrissis et al. [17] for SP 2.2. In addition, Persse [64] and O'Regan [112] indicated that performing revisions and ensuring up-to-date information are essential activities that are required to achieve and satisfy SP 2.2. The PPQA auditor/reviewer is required to ensure that the PPQA records established in the previous stage, "record," are kept up to date. If the status of PPQA activities is not found to be recent, then revisions are performed in order to run through the status of these activities, modify the revision history, and update the quality report. In other words, this stage aims to revise the PPQA records with the latest information so that they are always kept up to date.

The third stage is "Share." This has been included because Persse [64] highlighted the importance of sharing the quality report, including the PPQA records, with project management, team members, and organizational management in the implementation SP 2.2, as this results in the establishment of valuable and effective PPQA records. In addition, Chemuturi [104] indicated the need to share these quality assurance reports to establish a periodic audit process. It can be concluded from CMMI v1.3 specifications [65], Chrissis et al. [17], and O'Regan [112] that periodically publishing the quality report for relevant stakeholders is important in determining the status of the audit process and will be reflected in its results. The quality report is shared at predefined intervals with the managers and process engineering group at this stage. In other words, the latest versions of this report that include the independent PPQA records should be distributed among stakeholders. This report is usually generated periodically, for instance monthly, quarterly, or half-yearly. This stage can be conducted through a discussion or via e-mail with the quality report attached to the people concerned. It is then necessary to give them adequate time to review the report. This stage may be seen as a preparatory element for the next stage, "Improve."

"Improve" is the last stage in the proposed model. It is included because its functionality consists of an analysis of the quality of the reports, finding opportunities for improvement and presenting to management, as is pointed out in the following references cited in the literature: Persse [64], Chemuturi [104], and O'Regan [112]. This stage can also be considered to be a logical stage because it highlights the end objective of the recording of the results. Based on the established quality report, the SQA managers need to examine this report and other discussion forums that are used to identify issues, gaps, best practices, and trends as part of process improvement, and then implement them at the organizational level. All obtained suggestions and observations are recorded in detail and presented to management for approval. Based on the obtained approval, the process improvement initiatives and changes are institutionalized in an organization-wide manner.

6.5 Evaluation of the proposed models

6.5.1 Evaluation of the proposed models through an expert panel review process

A review process that involved a panel of experts was carried out to perform the evaluation of the models that we proposed. This process was used to get the opinions of six experts about the practice satisfaction, ease of learning and ease of use, user satisfaction, and applicability to small- and medium-sized software development organizations of the models that were proposed. The experts who took part in the evaluating process are listed in to show how they were identified.

Table 22 SPI experts' profile

SPI expert	Job title	Experience of SPI expert (years)	Knowledge of CMMI (Low 1 - 5 High)
1	Quality Officer/IT Auditor	10	4
2	Quality Officer/IT Auditor	12	4
3	Project manager	15	4
4	Project manager	13	4
5	Software developer	7	3
6	Software developer	8	4

It is vital to add at this point that there was one expert reviewer who was already known to us and who played an important role in pointing us to a number of the other experts that we used. A total of 10 letters were sent to these experts, asking them to take part in the evaluation. However, we only got six responses. We selected these experts because they all had sufficient experience in making improvements to the way in which software is produced.

We then decided to create a questionnaire to get the experts' views about the models. Some of the questions that we used were extracted from [93]. These questions were then made compatible with our research objectives. This was done by tailoring or customizing each of the questions. The questionnaire's three main sections included a cover letter, where the evaluation's objectives were explained; demographics; and model feedback. Its last section, which was the model feedback section, addressed four vital aspects: practice satisfaction, ease of learning and ease of use, user satisfaction, and applicability to small- and medium-sized software development organizations. Before the questionnaire was sent to the experts for evaluation, the questions were examined by an academic researcher. After he replied, a number of questions were rewritten to produce a clearer and more concise questionnaire. This also enabled us to capture more effectively the data that we needed.

The questionnaire comprised seven questions for each proposed model in the following categories (see Appendix):

- **Category 1:** 5-point scale of *strongly agree* to *strongly disagree* (three questions).

- **Category 2:** 5-point scale of *very* to *not at all* (three questions).
- **Category 3:** Open-ended question that asked the SPI expert if he or she would like to provide comments on how we could improve our proposed workflow model (one question).

A summary of the primary conclusions from the data collected is discussed below and presented in Table 23.

All of the expert reviewers who looked at ***practice satisfaction*** chose *strongly agree* or *agree* for every one of the models proposed. When the models proposed for SP1.1 and SP1.2 were considered, three of the experts selected *strongly agree*, and the other three chose *agree*. For the model proposed for SP 2.1, two of the experts selected *strongly agree*, and the remaining four chose *agree*. For the model proposed for SP 2.2, one of the experts selected *strongly agree*, and the other five selected *agree*. The results indicated that all of the models that we proposed met the CMMI v1.3 specifications because the question that was related to practice satisfaction had been created according to the aims of the four specific practices in CMMI v1.3's specifications.

We asked a couple of clearly defined questions about "ease of learning and ease of use" (i.e., RQ2) in the evaluation form for each specific practice. First, an inquiry was made concerning the clarity of the representation of the proposed model; second, a question was asked regarding how much knowledge was needed to use this model.

For the first question, all of the expert reviewers selected either 5 or 4 on a 5-point scale that ranged from *very clear* to *not at all clear* for the proposed models for SP 1.1 and SP 1.2. For the model proposed for SP 2.1, two of the expert reviewers selected 5, three selected 4, and one selected 3 on the same scale. Regarding the model that was proposed for SP 2.2, one of the expert reviewers selected 5, four selected 4, and one selected 3 on the same scale. These results revealed that the experts who took part in our evaluation stage had no problems with learning and understanding the proposed models. In addition, the workflow (the activities involved in every stage of the models that were proposed) was easily understood.

The experts were also asked to answer the second question by using a 1 to 5 scale, in which 5 is *too much knowledge* and 1 is *not at all*. Four experts chose 1 and two experts chose 2 for the models that were proposed for SP 1.1 and SP 1.2. For the model proposed for SP 2.1, three experts selected 1, and the other three selected 2 on the same scale. For the proposed model for SP 2.2, two experts selected 1, three selected 2, and one selected 3.

It can be concluded by studying the responses to the two questions about ease of learning and ease of use that, in general, the proposed models were not only clear, but also proved easy to understand, as not much CMMI knowledge was required to understand them. It is vital to add that the expert reviewers were able to understand the proposed models more quickly and easily because the models were split into a certain number of core stages, which was encouraging as it showed we had successfully designed both comprehensive and concise models.

The experts were asked two specific questions about each proposed model to help them evaluate the models against the stakeholder satisfaction criteria (i.e., RQ3). One pertained to how useful the model was in the software industry, and the other related to how the quality of software products could be improved as a result of the model. In other words, on each proposed model, the following two questions were asked:

- How useful would it be to the software industry to use our proposed workflow model (using a 5-point scale of 5 = *very useful* and 1 = *not at all*)?
- The use of our proposed workflow model would improve the software process and lead to the production of high-quality software products (using a 5-point scale of *strongly agree* to *strongly disagree*).

All the expert reviewers thought that the proposed models would be helpful to the software industry. When it came to the second question, for the proposed model for SP 1.1, three of the expert reviewers selected *strongly agree*, and the other three experts selected *agree*. For the model proposed for SP 1.2, two experts chose *strongly agree*, and the other four selected *agree*. For the proposed model for SP 2.1, two experts selected *strongly agree*, three selected *agree*, and one selected *neutral*. For the model proposed for SP 2.2, one expert selected *strongly agree*, four selected *agree*, and one selected *neutral*. These results showed that using the proposed models could improve the software process and that models would help to create a software product of high quality. There are, therefore, indications from the collected responses that the proposed models will make sure that “stakeholder satisfaction” exists.

We also examined the “applicability of the models to small- and medium-sized software development organizations” and asked the expert reviewers to rate every one of the proposed models using a 5-point scale (*strongly disagree* to *strongly agree*) to rate this statement: “Our proposed workflow model is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations” (i.e., RQ4).

For the proposed models for SP 1.1 and SP 1.2, all of the experts selected *strongly agree* or *agree*. For the model proposed for SP 2.1, two experts selected *strongly agree*, three selected *agree*, and one selected *neutral*. For the model proposed for SP 2.1, two experts selected *strongly agree*, three selected *agree*, and one selected *neutral*. This strongly indicates that the proposed models for SP 1.1, SP 1.2, SP 2.1, and SP 2.2 can be applied to small- and medium-sized software development organizations.

Evaluation criteria		Evaluation results							
		SP 1.1		**SP 1.2**		**SP 2.1**		**SP 2.2**	
		Number of Experts	Selection	Number of Experts	Selection	Number of Experts	Selection	Number of Experts	Selection
Practice satisfaction	The proposed model satisfied the goal of the specific practice, according to CMMI v1.3 specifications (strongly agree – strongly disagree)	(2) Senior Experts (1) Intermediate Expert	“Strongly agree”	(2) Senior Experts (1) Intermediate Expert	“Strongly agree”	(1) Senior Expert (1) Intermediate Expert	“Strongly agree”	(1) Senior Expert	“Strongly agree”
		(1) Intermediate Expert (2) Junior Experts	“Agree”	(1) Intermediate Expert (2) Junior Experts	“Agree”	(1) Senior Expert (1) Intermediate Expert (2) Junior Experts	“Agree”	(1) Senior Expert (2) Intermediate Experts (2) Junior Experts	“Agree”
Ease of learning and ease of use (*This provides the answer to RQ2*)	How clear is the representation of the proposed model? (Not at all 1 - 5 Very Clear)	(1) Senior Expert (1) Intermediate Expert	“5”	(1) Intermediate Expert	“5”	(1) Senior Expert (1) Intermediate Expert	“5”	(1) Intermediate Expert	“5”
		(1) Senior Expert (1) Intermediate Expert (2) Junior Experts	“4”	(2) Senior Experts (1) Intermediate Expert (2) Junior Experts	“4”	(1) Senior Expert (1) Intermediate Expert (1) Junior Expert	“4”	(2) Senior Experts (1) Intermediate Expert (1) Junior Expert	“4”
						(1) Junior Expert	“3”	(1) Junior Expert	“3”
	How much knowledge of CMMI is required to learn how to use the proposed model? (Not at all 1 - 5 Too much knowledge)	(1) Senior Expert (2) Intermediate Experts (1) Junior Expert	“1”	(2) Senior Experts (1) Intermediate Expert (1) Junior Expert	“1”	(1) Senior Expert (2) Intermediate Experts	“1”	(1) Senior Expert (1) Intermediate Expert	“1”
		(1) Senior Expert (1) Junior Expert	“2”	(1) Intermediate Experts (1) Junior Expert	“2”	(1) Senior Expert (2) Junior Experts	“2”	(1) Senior Expert (1) Intermediate Experts (1) Junior Expert	“2”
								(1) Junior Expert	“3”
User satisfaction (*This provides the answer to RQ3*)	How useful would it be to the software industry to use the proposed model? (Not at all 1 - 5 Very useful)	(1) Senior Expert (1) Intermediate Expert	“5”	(1) Senior Expert (1) Intermediate Expert	“5”	(1) Senior Expert	“5”	(2) Senior Experts (2) Intermediate Expert (2) Junior Expert	“4”
		(1) Senior Experts (1) Intermediate Expert (2) Junior Expert	“4”	(1) Senior Experts (1) Intermediate Expert (2) Junior Expert	“4”	(1) Senior Expert (2) Intermediate Expert (2) Junior Expert	“4”		
	The use of the proposed model would improve the software process and lead to the production of high quality products (strongly agree – strongly disagree)	(1) Senior Experts (2) Intermediate Experts	“Strongly agree”	(1) Senior Expert (1) Intermediate Expert	“Strongly agree”	(1) Senior Expert (1) Intermediate Expert	“Strongly agree”	(1) Intermediate Expert	“Strongly agree”
		(1) Senior Experts (2) Junior Expert	“Agree”	(1) Senior Experts (1) Intermediate Expert (2) Junior Expert	“Agree”	(1) Senior Expert (1) Intermediate Expert (1) Junior Expert	“Agree”	(1) Senior Expert (2) Intermediate Experts (1) Junior Expert	“Agree”
						(1) Junior Expert	“Neutral”	(1) Junior Expert	“Neutral”
Applicability of the models to small- and medium-sized	The proposed model is applicable to small- and medium-sized software	(1) Senior Experts (1) Intermediate Expert	“Strongly agree”	(1) Senior Expert (1) Junior Expert	“Strongly agree”	(1) Senior Expert (1) Intermediate Experts	“Strongly agree”	(1) Intermediate Expert	“Strongly agree”

software organizations (*This provides the answer to RQ4*)	development organizations. (strongly agree – strongly disagree)	(1) Senior Experts (1) Intermediate Expert (2) Junior Experts	"Agree"	(1) Senior Experts (2) Intermediate Expert (1) Junior Expert	"Agree"	(1) Senior Expert (1) Intermediate Expert (1) Junior Expert	"Agree"	(2) Senior Expert (1) Intermediate Experts	"Agree"
						(1) Junior Expert	"Neutral"	(2) Junior Experts	"Neutral"

Table 23: Summary of evaluation result

6.5.2 Novelties of the proposed models

The primary novelties of the present study are highlighted as follows: First, we proposed high-level models for the specific practices of the PPQA process area. These proposed models meet the CMMI v1.3 specifications and were designed according to the objectives of each specific practice stated in the CMMI v1.3 specifications. Second, the PPQA process area was addressed at the specific practice level. Third, the proposed models for SP 1.1, SP 1.2, SP 2.1, and SP 2.2 were developed mainly for small- and medium-sized software development organizations.

Moreover, the evaluation of the proposed models regarding practice satisfaction, ease of learning and ease of use, user satisfaction, and applicability to small- and medium-sized software development organizations was provided.

Another major innovation of this study is that the guidelines are presented. These include the templates, checklists, and forms that users can employ to traverse our proposed models. This was performed because O'Regan [27] said, "Templates support the process and allow consistent input and output during the different parts of the process." Moreover, the author indicated that templates and checklists help ensure that the activities associated with process areas are completed. Afrooz [94] also said, "The template would provide the organization with a guideline to achieve the organizational objectives." Furthermore, [95] provides an example that presents how a specific practice in CMMI can be implemented by proposing templates, checklists, and forms.

6.6 Remarks

The most important objective of the chapter was to create a high-level model for each specific practice in the PPQA at CMMI Maturity Level 2. These models are specifically for small- and medium-sized software development organizations. The objective also included evaluating the proposed models by using a process involving a review panel of experts, so RQ1, RQ2, RQ3, and RQ4 were all addressed for this purpose.

We developed a workflow model for each specific practice in PPQA to deal with RQ1 and also identified a list of the guidelines and satisfactory models from the literature on the four specified practices. A significant number of research articles and case studies in the field of SPI were explored during the literature review, including such SPI models as CMMI and software requirements engineering. We also carried out an evaluation of the data that were collected from the literature review to come up with the proposed models. It is important to add that proposed comments and suggestions by expert reviewers were employed to advance the development of the proposed model.

The evaluation of the proposed workflow models was carried out through the use of an expert review panel to tackle RQ2, RQ3, and RQ4. For RQ2, the expert reviewers were given specific questions about how clearly the proposed models were represented and the knowledge range needed to use them successfully.

Two specific questions were then clearly defined regarding RQ3 in the evaluation form. This was done to assist in the evaluation of the proposed models against the criteria of stakeholder satisfaction. The first of these related to how useful the proposed models were overall in the software industry, and the second regarded what improvements could be gained through the adoption of the proposed models. The experts were asked a particular question for RQ4 about the applicability of the proposed models to small- and medium-sized software development organizations.

In short, the overall evaluation results revealed that the proposed models were designed to ensure practice satisfaction, according to the requirements of CMMI Maturity Level 2. The results were also able to satisfy the ease of learning and ease of use criteria. They, therefore, proved be clear and easy to both learn and use. In addition, they met the stakeholders' expectations and the desired level of satisfaction. Moreover, small- and medium-sized software development organizations are capable of adopting these proposed models into their own environments.

CHAPTER SEVEN: CONCLUSIONS

7.1 Summary of the problem statement

A lot of researchers have placed emphasis on the quality of software through the application of software process improvements (SPI) (Ashrafi [4] ; García-Mireles et al. [5]). The impact of SPI methodologies on software quality has been investigated by Ashrafi [4], and García-Mireles et al. [5] state that software development organisations have been seeking to enhance the quality of the resulting software through the implementation of SPI, which is seen as a vital part of the optimisation of the software development process, particularly for small- and medium-sized organisations (Iqbal et al. [6]; Rahmani et al.[7]).

Niazi et al. [8] have said the development of effective SPI implementation design initiatives to aid small- and medium-sized organisations are a major challenge for the software industry. Thus, researchers have made it their main priority to implement he SPI frameworks and standards in order to improve both the software's quality and the rate of productivity achieved (Dutra et al. [9]; Niazi [10]). The latest SPI model to be introduced by the Software Engineering Institute (SEI) is CMMI, which helps software development organisations to improve the quality of their software. "CMMI has been widely researched," according to Lee et al. [11].

Staples et al. [13] believe that the most important reasons why the majority of software organizations have not used it as an SPI model are the organizations are too small, it takes a lengthy time to adopt this kind of model, the expensive services of capability maturity, and the use of other SPI models. Furthermore, Niazi et al. [14] have highlighted the two main reasons why a lot of software organizations do not have the will to begin the lengthy path of implementing CMMI – the major investment that is needed and small amount of success it achieves.

Chrissis et al. [17] wrote that: "CMMI identifies 'what' activities are expected, but does not specify techniques on how to accomplish those activities." In addition, Vivatanavorasin et al. [18] highlighted that CMMI does not give any details about what actions these software development organisations should take in order to achieve a given CMMI process area. As a result, no clear approaches are available that can help small- and medium-sized software companies to implement CMMI. Greater attention on exactly "how" to implement CMMI is required to help these organisations to adopt different CMMI Levels successfully.

Niazi et al. [8] highlighted that small- and medium-sized software organisations aim to achieve Level 2 once they start to accept CMMI as an SPI model, so CMMI Level 2 was therefore be particularly addressed in this proposed research. The objective of this research work was to implement the REQM and PPQA process area of CMMI for small- and medium-sized software development organizations.

7.2 Summary of research methodology

Setting criteria for successful outcomes was the first stage involved in the design of the proposed models. The following basic criteria were used for building the models

- Ease of learning and ease of use,
- user satisfaction
- Practice satisfaction
- Applicability to small- and medium-sized software development organizations

It is important to stress that the primary reason for carefully selecting the above criteria comes from the findings presented in Niazi et al. [118] and Niazi et al. [119]. Moreover, these criteria have been specified as a result of the references reported in the literature from Niazi et al. [80] and Niazi et al. [87].

To address the desired criteria, research questions were developed in the second stage. In the third stage, previous research that targets the implementation of CMMI Level 2 process was reviewed to answer the research questions. During this stage, published research articles that offered recommendations about implementing CMMI Level 2 were explored. In the fourth stage, a very careful analysis and evaluation of the collected data was carried out. The proposed models for the specific practices in the REQM and PPQA process area were developed in stage 5. In stage 6, the evaluation of the proposed models was performed using an expert panel review process.

7.3 Critique of methodology

There are a number of distinct limitations pertaining to this research study. Of the limitations, the most crucial ones are noted below:

1. The SPI expert reviewers might interpret each question differently when responding to the questionnaire and, therefore, answer based on their own understanding of the question.
2. The SPI expert reviewers might interpret the descriptions of the proposed models, appended with the questionnaire, differently.

It is worth noting here that there is no discernible evidence about the previous two limitations actually being experienced, as none of the expert reviewers have reported and expressed any issues or concerns. However, in order to lessen the potential impact of the two limitations, the questionnaire, including the descriptions for the proposed models, was examined by an academic researcher before the experts were asked to fill it out. The final version of the questionnaire was developed based on the researcher's feedback. This was done to make it clearer to respondents, thus reducing the possibility that the questionnaire, as well as the descriptions, might be interpreted differently. Moreover, the proposed models in this study were based on a detailed literature review, where published research articles that provided guidelines for implementing CMMI Level 2 were analyzed.

3. The questionnaire in the evaluation stage consisted of six closed-ended questions about each of the proposed models, which did not offer the full flexibility the expert reviewers needed to express and explain their own feelings and opinions about the proposed models.

However, we lessened the potential impact of this limitation by including one open-ended question. This asked expert reviewers if he/she would like to provide comments on how we could improve the proposed models. This question gave the expert reviewers the chance to respond freely and include more information.

4. The knowledge of the expert reviewers as well as their experiences might place constraints on our evaluation results.

Despite this limitation, we are confident about the evaluation results obtained, as all expert reviewers involved in the evaluation process have sufficient knowledge in the field of SPI and RE. In addition, there were no suggestions put forward by the researcher to explore these expert reviewers.

7.4 Summary of findings

There is a significant need to give careful consideration to REQM and PPQA of CMMI Level 2, especially in the context of small- and medium-sized software development organizations, in order to assist such organizations in getting one-step closer to achieving CMMI Level 2 certification. The objective of this research is to report on the implementation of REQM and PPQA process areas, especially for small- and medium-sized software development organizations. In this paper, we propose an abstract-level model for each specific practice of the REQM and PPQA as well as cover the evaluation of the models. The proposed models are based on a significant amount of research in software process improvement (SPI), CMMI, and requirements engineering.

In order to address the RQ1, we developed a workflow model for each specific practice of REQM and PPQA. For each specific practice, we identified the guidelines and suitable models from the literature and believe that our findings can be utilized to implement each specific practice of REQM and PPQA. During the literature review, we collected data by exploring published research articles, experience reports, and case studies. The development of the proposed workflow model for each specific practice is based mainly on our findings from the literature.

In order to address the RQ2, RQ3, and RQ4, we performed the evaluation of the proposed models by using the expert review process. The evaluation results show that our proposed models are clear, easy to use, and easy to learn. In addition, our models can be applied to both small- and medium-sized software development organizations. Moreover, they can help such organizations implement REQM and PPQA process areas according to the CMMI Level 2 maturity requirement.

This study will help small- and medium-sized software organizations adopt CMMI Level 2 specific practices of the REQM and PPQA process areas quickly. Thus, it will assist them in getting one step closer to achieving CMMI Level 2 certification. In addition, this will lead to high-quality products and customer satisfaction.

Furthermore, our implementation will increase the depth of knowledge of both practitioners and researchers on each specific practice of the REQM and PPQA process areas at CMMI Maturity Level 2. Moreover, we can consider this study as an ongoing research work that can be extended and carried out by researchers in the future. It can possibly lead to research expansion in the field of CMMI Level 2 implementation.

For future work, we intend to evaluate the proposed workflow models by conducting multiple case studies in an industrial setting in order to assess their suitability.

7.5 Reflection on research

I believe that the most important part of research is discovering things that you did not know about before. I have found that utilizing various techniques, methods, and procedures enables one to find out something new. So I set about learning the research process, including procedures like research design, conducting data collection, providing data analysis, and writing reports. I learned well how to carry out a survey in order to investigate a research question that is specific to a certain kind of literature. It is clear from this cyclic process that a researcher is not someone who knows all the correct answers, but instead, he/she is someone that works hard to find the correct research questions.

7.6 Research Outcomes

This research highlighted the following useful outcomes**:**

- High-level (abstract-level) Models for the specific practices of Requirements Management (REQM) of CMMI Level 2.
- High-level (abstract-level) models for the specific practices of the Process and Product Quality Assurance (PPQA) of CMMI Level 2.
- Identifying PPQA, REQM CMMI Level 2 best practices.
- Guidelines, templates, forms, and checklists for the most important processes in the developed models for specific practices of the Process and Product Quality Assurance (PPQA) and Requirements Management (REQM).
- Evaluation of the models.

End

REFERENCES

[1]. Halvorsen, Christian Printzell, and Reidar Conradi. "A taxonomy to compare SPI frameworks." Software Process Technology. Springer Berlin Heidelberg, 2001. 217-235.

[2]. B. Kitchenham and S. Pfleeger, “Software Quality: The Elusive Target,” IEEE Software, vol. 13, no. 1, pp. 12-21, Jan. 1996.

[3]. W. Scacchi, “Process Models in Software Engineering,” Encyclopedia of Software Eng., pp. 993-1005, 2001.

[4]. Ashrafi, Noushin. "The impact of software process improvement on quality: in theory and practice." Information & Management 40.7 (2003): 677-690

[5]. García-Mireles, Gabriel Alberto, et al. "The Influence of Process Quality on Product Usability: A Systematic Review." CLEI Electronic Journal 16.2 (2013): 6-6.

[6]. Iqbal, J., Ahmad, R. B., Nasir, M. H. N. M., Niazi, M., Shamshirband, S. and Noor, M. A. “Software SMEs’ unofficial readiness for CMMI®-based software process improvement”, Software Quality Journal (2015): 1-27

[7]. Rahmani, Hosein, Ashkan Sami, and Abdullah Khalili. "CIP-UQIM: A unified model for quality improvement in software SME's based on CMMI level 2 and 3." Information and Software Technology 71 (2016): 27-57.

[8]. Niazi, M. and Babar, M. 2009. Identifying High Perceived Value Practices of CMMI Level 2: An Empirical Study, Information and Software Technology Journal 51 (8). 1231-1243.

[9]. Dutra, Eliezer, and Gleison Santos. "Software Process Improvement Implementation Risks: A Qualitative Study Based on Software Development Maturity Models Implementations in Brazil." Product-Focused Software Process Improvement. Springer International Publishing, 2015. 43-60.

[10]. Niazi, Mahmood. "A comparative study of software process improvement implementation success factors." Journal of Software: Evolution and Process 27.9 (2015): 700-722.

[11]. Lee, Chang-Shing, and Mei-Hui Wang. "Ontology-based computational intelligent multi-agent and its application to CMMI assessment." Applied Intelligence 30, no. 3 (2009): 203-219.

[12]. Day, Bryce, Sean Chin Ke-Zun, Levi Lovelock, and Christof Lutteroth. "Climbing the ladder: CMMI level 3." In Enterprise Distributed Object Computing Conference, 2009. EDOC'09. IEEE International, pp. 97-106. IEEE, 2009.

[13]. Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P. and Murphy, R. 2007. An Exploratory Study of Why Organizations Do Not Adopt CMMI, Journalof Systems and Software. 80 (6). 883-895.

[14]. Niazi, M. and Babar, M., Ali. 2007. De-motivators for software process improvement :An Analysis of Vietnamese Practitioners' Views. International Conference on Product Focused Software Process Improvement PROFES 2007, LNCS 4589. 118-131.

[15]. Al-Tarawneh, Mejhem Yousef, Mohd Syazwan Abdullah, and Jasem Alostad. "Software Development Process Improvement Framework (SDPIF) for Small Software Development Firms (SSDFs)." International Journal of Computer Science Issues (IJCSI) 10.1 (2013).

[16]. Clarke, Paul, and Rory V. O'Connor. "An empirical examination of the extent of software process improvement in software SMEs." Journal of Software: Evolution and Process 25, no. 9 (2013): 981-998

[17]. Mary Chrissis, Mike Konrad, Sandra Shrum, “CMMI®: Guidelines for Process Integration and Product Improvement”, 2nd Edition, Pearson Education, Nov 2006.

[18]. Vivatanavorasin, Chakkanart, Nakornthip Prompoon, and Athasit Surarerks. "A Process Model Design and Tool Development for Supplier Agreement

Management of CMMI: Capability Level 2." In Software Engineering Conference, 2006. APSEC 2006. 13th Asia Pacific, pp. 385-392. IEEE, 2006.

[19]. K. El Emam, H.N. Madhavji, A field study of requirements engineering practices in information systems development, in: Second International Symposium on Requirements Engineering, 1995, pp. 68–80.

[20]. Standish-Group: Chaos: The Standish Group Report. Standish Group International (1995)

[21]. Tejas Shah, S V Patel, "A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods", International Journal of Computer Applications (0975 – 8887) Volume 99 –No.15, August 2014.

[22]. Pacheco, C. L., Garcia, I. A., Calvo-Manzano, J. A., & Arcilla, M. (2015). A proposed model for reuse of software requirements in requirements catalog.Journal of Software: Evolution and Process, 27(1), 1-21.

[23]. Pratt, David. The IT Project Management Answer Book. Do Sustainability, 2012.

[24]. Young, Ralph Rowland, Steven M. Brady, and Dennis C. Nagle. How to save a failing project: chaos to control. Management Concepts Inc., 2009.

[25]. A. Jarvis, V. Crandall, INROADS to Software Quality, Prentice-Hall Inc., 1997.ISBN: 0132384035.

[26]. Niazi, Mahmood, Muhammad Ali Babar, and Suhaimi Ibrahim. "An empirical study identifying high perceived value practices of CMMI level 2." In Product-Focused Software Process Improvement, pp. 427-441. Springer Berlin Heidelberg, 2008.

[27]. Nasir, Mohd Hairul Nizam Md, Rodina Ahmad, and Noor Hafizah Hassan. "Resistance factors in the implementation of software process improvement project." In Information Technology, 2008. ITSim 2008. International Symposium on, vol. 4, pp. 1-10. IEEE, 2008.

[28]. Niazi, Mahmood. "Software process improvement implementation: avoiding critical barriers." CROSSTALK. The Journal of Defense Software Engineering22, no. 1 (2009): 24-27.

[29]. Unterkalmsteiner, Michael, Tony Gorschek, AKM Moinul Islam, Chow Kian Cheng, Rahadian Bayu Permadi, and Robert Feldt. "Evaluation and measurement of software process improvement—A systematic literature review." Software Engineering, IEEE Transactions on 38, no. 2 (2012): 398-424.

[30]. I. Sommerville, Software Engineering, eighth ed. Pearson Education,2004.

[31]. G. Cugola and C. Ghezzi, “Software Processes: A Retrospective and a Path to the Future,” Software Process: Improvement and Practice, vol. 4, no. 3, pp. 101-123, Sept. 1998.

[32]. M.C. Paulk, W.S. Humphrey, G.J. Pandelios, Software Process Assessments: Issues and Lessons Learned, Proceedings of ISQE92, pp. 4B41-4B513 ,1992

[33]. C. Fox, W. Frakes, The Quality Approach: Is It Delivering?, Communications of the ACM, Vol. 40 ,No.6, pp.25-29, June, 1997.

[34]. Niazi, M. and Babar, M. 2009. Identifying High Perceived Value Practices of CMMI Level 2: An Empirical Study, Information and Software Technology Journal 51 (8). 1231-1243.

[35]. Moser, Thomas, et al. "Requirements management with semantic technology: an empirical study on automated requirements categorization and conflict analysis." Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2011

[36]. Monteiro, Paula, Ricardo Jorge Machado, and Rick Kazman. "Inception of Software Validation and Verification Practices within CMMI Level 2." Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on. IEEE, 2009.

[37]. Hwang, Sun Myung, and Hee Gyun Yeom. "Analysis of Relationship among ISO/IEC 15504, CMMI and K-model." In Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD'09. 10th ACIS International Conference on, pp. 306-309. IEEE, 2009.

[38]. "Leveraging Application Lifecycle Management to Achieve CMMI Maturity Level 3 and Higher on Government Program", Parametric Technology Corporation (PTC), 2011.

[39]. R. W. Reitzig; John B. Miller; Dave West; Raymond L. Kile. "Achieving Capability Maturity Model Integration Maturity Level 2 Using IBM Rational Software's Solutions." Rational White Paper, 2003.

[40]. Crespo, Daniel, and Mercedes Ruiz. "Decision making support in CMMI process areas using multiparadigm simulation modeling." In Simulation Conference (WSC), Proceedings of the 2012 Winter, pp. 1-12. IEEE, 2012.

[41]. McCabe, T.J.: A Complexity Measure. IEEE Transactions on Software Engineering 2, 308–320 (1976)

[42]. Batista, J. and Dias, d. F .2000 .Software Process Improvement in a Very Small Team: a Case with CMM, Software Process-Improvement and Practice (5). 243-250

[43]. Eileen Forrester, Brandon L. Buteau, Sandra Shrum, Introduction to CMMI for Services: Guidelines for Superior Service, 2nd Edition, Published by Addison-Wesley Professional. Part of the SEI Series in Software Engineering series, March, 2011.

[44]. Xu, Gang, et al. "Supporting Flexibility of the CMMI Process Framework with a Multi-layered Process Model." Web Information System and Application Conference (WISA), 2013 10th. IEEE, 2013.

[45]. Haron, A., Harun, M., Naz'ri Mahrim, M., Sahibuddin, S., Zakaria, N. H., & Abdul Rahman, N. (2012, April). Understanding the requirement engineering for organization: the challenges. In Computing Technology and Information Management (ICCM), 2012 8th International Conference on (Vol. 2, pp. 561-567). IEEE.

[46]. Saeed, Saqib, Ashi Iram, Kiran Nazeer and Tayyaba Ayub. "Understanding Requirement Engineering Practices: An Empirical Study of the Pakistani Software Industry." Knowledge-Based Processes in Software Development. IGI Global, 2013.

[47]. Rickman, Dale M. "A new process for requirements understanding." Digital Avionics Systems, 2001. DASC. 20th Conference. Vol. 1. IEEE, 2001.

[48]. Kotonya, G. and Sommerville, I. (1998): Requirements Engineering – Processes and
Techniques, John Wiley & Sons, UK.

[49]. I. Sommerville, Software Engineering, 9th ed. Pearson Education,2011

[50]. Grady, Jeffrey O. System engineering planning and enterprise identity. Vol. 7. CRC Press, 1995.

[51]. Leffingwell, Dean. Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Addison-Wesley Professional, 2010.

[52]. Verner, June, et al. "Requirements engineering and software project success: an industrial survey in Australia and the US." Australasian Journal of Information Systems 13.1 (2007).

[53]. Colin Hood CMMI: What do we need to do in Requirements Management & Engineering? HOOD Group, February 2003

[54]. West, Michael. Real process improvement using the CMMI. CRC Press, 2004.

[55]. Kasse, Tim. Practical insight into CMMI. Artech House, 2008

[56]. Ing, David. "Offerings as Commitments and Context: Service Systems from a Language Action Perspective." 154-172.

[57]. Tariq, Anum, Shoab Ahmed Khan, and Sundas Iftikhar. "Remapping of CMMI level-2 KPA's for development process improvement of Software-as-a-Service (SaaS) cloud environment." Open Source Systems and Technologies (ICOSST), 2014 International Conference on. IEEE, 2014.

[58]. Morris, Jonathan, and Farid Baddache. "Back to Basics: How to Make Stakeholder Engagement Meaningful for Your Company." The Business of a Better World (2012).

[59]. Business Strategy: an Introduction: David Campbell, George Stonehouse and Bill Houston, Butterworth Heinemann (2003, 2nd edition)

[60]. Schmeer, Kammi. "Stakeholder analysis guidelines." Policy toolkit for strengthening health sector reform (1999): 1-33.

[61]. G. O'Regan, Introduction to Software Process Improvement, Undergraduate Topics in Computer Science, Springer-Verlag London Limited 2011

[62]. Wilkie, F. George, Donald McFall, and Fergal McCaffery. "An evaluation of CMMI process areas for small to medium sized software development organizations." Software Process: Improvement and Practice 10.2 (2005): 189-201.

[63]. Lester, N. G., Wilkie, F. G., McFall, D., Ware, M. P. Investigating the role of CMMI with expanding company size for small- to medium-sized enterprises. Journal of Software: Evolution and Process 22(1), 17-31, (2010).

[64]. James Persse, "Project Management Success with CMMI: Seven CMMI Process Areas", Pearson Education, 2007

[65]. http://www.sei.cmu.edu/reports/10tr033.pdf

[66]. E. J. Barry, T. Mukhopadhyay, and S. A. Slaughter, "Software Project Duration and Effort: An Empirical Study," Information Technology and Management, vol. 3, 1-2, pp. 113-136, 2002.

[67]. McGee, S. and Greer, D. (2009) A Software Requirements Change Source Taxonomy. 2009 4th International Conference on Software Engineering Advances, Porto, 20-25 September 2009, 51-58.

[68]. Pierce, K., Ingram, C., Bos, B. and Ribeiro, A. (2013) Experience in Managing Requirements between Distributed Parties in a Research Project Context. 2013 IEEE 8th International Conference on Global Software Engineering (ICGSE), Bari, 26-29 August 2013, 124-128.

[69]. B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," Comunications of the ACM, vol. 31, 11, pp. 1268-1287, 1988.

[70]. Nurmuliani, Nur, Didar Zowghi, and S. Powell. "Analysis of requirements volatility during software development life cycle." Software Engineering Conference, 2004. Proceedings. 2004 Australian. IEEE, 2004.

[71]. Zowghi, D., Nurmuliani, N.: A study of the impact of requirements volatility on software project performance. In: Ninth Asia-Pacific Software Engineering Conference, pp. 3–11 (2002)

[72]. Manifesto, C. H. A. O. S. "Think Big, Act Small." The Standish Group International Inc (2013).

[73]. M. Niazi, An empirical study for the improvement of requirements engineering process, in: The 17th International Conference on Software Engineering and Knowledge Engineering, July 14–16, 2005, Taipei, Taiwan, Republic of China, 2005, pp. 396–399.

[74]. M. Niazi, S. Shastry, Role of requirements engineering in software development process: an empirical study, in: IEEE International Multi-Topic Conference (INMIC03), 2003, pp. 402–407.

[75]. Gorschek, T., May 2006. Requirements engineering supporting technical product management. Ph.D. thesis, Dept. of Systems and Software Engineering, Blekinge Institute of Technology.

[76]. Gorschek, T., Wohlin, C., 2005. Requirements abstraction model. Requirements Engineering 11 (1), 79–101.

[77]. Ramesh, B., 1998. Factors influencing requirements traceability practice. Communications of the ACM 41 (12), 37–44.

[78]. Sommerville, I., Sawyer, P., 1997. Requirements engineering: A good practice guide. John Wiley & Sons, Inc., New York, NY, USA.

[79]. Abran, A., Moore, J. W., Bourque, P., Dupuis, R. (Eds.), 2004. Guide to the software engineering body of knowledge (SWEBOK). IEEE Computer Society, Los Alamitos, California.

[80]. Niazi, M., Wilson, D., Zowghi, D.: A Maturity Model for the Implementation of SoftwareProcess Improvement: An empirical study. Journal of Systems and Software 74(2), 155–172 (2005)

[81]. Vishnuvarthanan Moorthy, “CMMI Implementation Guide: A Practitioner’s Perspective”, 2013.

[82]. West, Michael. Real process improvement using the CMMI. CRC Press, 2004.

[83]. Tariq, Anum, Shoab Ahmed Khan, and Sundas Iftikhar. "Remapping of CMMI level-2 KPA's for development process improvement of Software-as-a-Service (SaaS) cloud environment." Open Source Systems and Technologies (ICOSST), 2014 International Conference on. IEEE, 2014.

[84]. Mäkäräinen, M.: Application management requirements for embedded software. Technical Research Centre of Finland, VTT Publications 286 (1996)

[85]. Olsen, N.: The software rush hour. IEEE Software, 29–37 (September 1993)

[86]. Ince, D.: Introduction to software quality assurance and its implementation. McGraw-Hill,New York (1994)

[87]. Niazi, M., Hickman, C., Ahmad, R., & Babar, M. A. (2008). A model for requirements change management: Implementation of CMMI level 2 specific practice. In Product-Focused Software Process Improvement (pp. 143-157). Springer Berlin Heidelberg.

[88]. Wasim Bhatti, M., Hayat, F., Ehsan, N., Ahmed, S., Ishaque, A., & Mirza, E. (2010, October). A methodology to manage the changing requirements of a software project. In Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on (pp. 319-322). IEEE.

[89]. Ramesh, Balasubramaniam, and Matthias Jarke. "Toward reference models for requirements traceability." Software Engineering, IEEE Transactions on 27.1 (2001): 58-93.

[90]. Kirova, V., Kirby, N., Kothari, D., & Childress, G. (2008). Effective requirements traceability: Models, tools, and practices. Bell Labs Technical Journal, 12(4), 143-157.

[91]. Azmi, Azri, and Suhaimi Ibrahim. "Test management traceability model to support software testing documentation." Digital Information and Communication Technology and Its Applications. Springer Berlin Heidelberg, 2011. 21-32.

[92]. Jonasson, H. (2012). Determining Project Requirements: Mastering the BABOK® and the CBAP® Exam. CRC Press.

[93]. Niazi, M. (2004). A framework for assisting the design of effective software process improvement implementation strategies. Ph.D. thesis, University of Technology Sydney.

[94]. Afrooz Ghalambor, “Designing a Process Measurement Program, as a part of Measurement & Analysis Process Area of CMMI Level 2”. Master Thesis, Borås University, School of Engineering, September 2009

[95]. “A CMMI Implementation Case Study”, InterGlobe Consulting

[96]. Khraiwesh, Mahmoud. "PROCESS AND PRODUCT QUALITY ASSURANCE MEASURES IN CMMI." International Journal of Computer Science & Engineering Survey 5, no. 3 (2014).

[97]. “Performance audit guide”, republic of KOSOVO office of the auditor general, august 2009.

[98]. Tom Gilchrist. “Software Process Reviews/Audits”, Seattle Area Software Quality Assurance Group-SASQAG (2002)

[99]. FILIP, FLORINA-CRISTINA, and V. L. A. D. I. M. I. R. MARASCU-KLEIN. "Analysis of Process and Product Quality Assurance."

[100]. Pankaj Jalote, “Software Project Management in Practice”, Addison-Wesley Professional, 2002

[101]. Westfall, Linda. The certified software quality engineer handbook. ASQ Quality Press, 2008.

[102]. Rohit Khurana, “Software Engineering - Principles and Practices”, Vikas Publishing House Pvt Ltd, Nov 1, 2009

[103]. IEEE Standard for Software Quality Assurance Processes Sponsor Software & Systems Engineering Standards Committee of the IEEE Computer Society Approved 27 March 2014 IEEE-SA Standards Board

[104]. Chemuturi, Murali. Mastering software quality assurance: best practices, tools and techniques for software developers. J. Ross Publishing, 2010.

[105]. Team, SCAMPI Upgrade. "Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document." (2011).

[106]. Jérôme Dern, “Embedded Software: Striving for excellence in development,” BoD - Books on Demand, (2015)

[107]. Kenett, Ron S., and Emanuel Baker. Process Improvement and CMMI® for Systems and Software. CRC Press, 2010.

[108]. Koirala, Shivprasad, and Sham Sheikh. Software testing. Jones & Bartlett Learning, 2009.

[109]. Sharma, Meena, and Rajeev Vishwakarma. "CMMI Based Software Metrics For OOAD." International Journal of Programming Languages and Applications (IJPLA) , Vol.3, No.1, January 2013.

[110]. Sharma, Meena, and Rajeev G. Vishwakarma. "Evaluation & Validation of Work Products in Unified Software Development Process." International Journal of Software Engineering & Applications 3.2 (2012): 97.

[111]. Satrio, Santoso, and Gaol Ford Lumban. "Measurement of Capability Maturity Model Integration Implementation Impact for Application Development Process Improvement in Telkomsigma." International Journal of Software Engineering and Its Applications 8.9 (2014): 103-124.

[112]. O'Regan, Gerard. Introduction to Software Quality. Undergraduate Topics in Computer Science, Springer 2014, ISBN 978-3-319-06105-4, pp. 1-341

[113]. Fahad, H. Alshammari, and Ahmad Rodina. "An empirical study on factors that affect the transition time between capability maturity model integration (CMMI) levels in Saudi Arabia." African Journal of Business Management 5, no. 17 (2011): 7342-7349.

[114]. Chen, Xi, Mark Staples, and Paul Bannerman. "Analysis of CMMI Specific Practice Dependencies." European Software Process Improvement and Innovation (EuroSPI 2008)} (2008): 94-105.

[115]. Khraiwesh, Mahmoud. "Project Planning Measures in CMMI." International Journal of Software Engineering & Applications (IJSEA) 4, no. 2 (2013): 103-121.

[116]. Khraiwesh, Mahmoud. "PROCESS AND PRODUCT QUALITY ASSURANCE MEASURES IN CMMI." International Journal of Computer Science & Engineering Survey 5, no. 3 (2014).

[117]. Yucalar, Fatih, and Senol Zafer Erdogan. "A questionnaire based method for cmmi level 2 maturity assessment." Journal of aeronautics and space technologies 4, no. 2 (2009): 39-46.

[118]. Niazi, M., Cox, K., & Verner, J. (2005). An empirical study identifying high perceived value requirements engineering practices. In Fourteenth International

Conference on Information Systems Development (ISD'2005). Karlstad University, Sweden August 15–17.

[119]. Niazi, M., & Shastry, S. (2003). Role of requirements engineering in software development process: An empirical study. In IEEE International Multi-Topic Conference (INMIC03) (pp. 402–407).

[120]. Ali Babar, Muhammad, and Barbara Kitchenham. "Assessment of a framework for comparing software architecture analysis methods." (2007).

[121]. Matthias R, Tools & Techniques Expert Opinion, Online available http://www.idemployee.id.tue.nl/g.w.m.rauterberg/ lecturenotes/UFTexpertopinion.pdf

[122]. Maa, Nan, Tracy Hallb, and Trevor Barker. "USING AN EXPERT PANEL TO EMPIRICALLY VALIDATE A REQUIREMENTS ENGINEERING MEDIATION MODEL." (2008).

[123]. Dybå T., (2000) An instrument for measuring the key factors of success in software process improvement, Empirical Software Engineering 5 (2000) (4), pp. 357–390.

[124]. HAKIM, C. ―Research Design: Strategies and Choices in the Design of Social Research. Contemporary Social Research: 13, ed. M. Bulmer, London: Routledge, 1987.

[125]. BEECHAM, S., HALL, T., BRITTON, C., COTTEE, M. & RAINER, A. Using an expert panel to validate a requirements process improvement model. Journal of Systems and Software, v. 76, p. 251-275, 2005.

[126]. DYBÅ, T. An Instrument For Measuring The Key Factors Of Success in Software Process Improvement. Empirical Software Engineering. 5 (4): p. 357-390, 2000.

[127]. EL EMAM, K. AND N.H. MADHAVJI. An Instrument for Measuring the Success of the Requirements Engineering Process in Information Systems Development. Empirical Software Engineering. v.1: p. 201-240, 1996.

[128]. Lauesen, Soren, and Otto Vinter. "Preventing requirement defects: An experiment in process improvement." Requirements Engineering 6.1 (2001): 37-50.

[129]. ROSQVIST, T., M. KOSKELO, AND H. HARJU. Software Quality Evaluation Based on Expert Judgement. Software Quality Journal. 11 (1): p. 39-55, 2003.

[130]. KITCHENHAM, B., S.L. PFLEEGER, B. MCCOLL, AND S. EAGAN, 2002. An empirical study of maintenance and development estimation accuracy. Journal of Systems and Software. 64 (1): p. 57-77, 2002.

[131]. COOKE, R. M. Experts in Uncertainty: Opinion and Subjective Probability in Science. Oxford University Press, USA, 1991.

[132]. Li, Ming, and Carol S. Smidts. "A ranking of software engineering measures based on expert opinion." Software Engineering, IEEE Transactions on 29.9 (2003): 811-824.

[133]. Khan, Samee U., Mahmood Niazi, and Rabiah Ahmad. "Empirical investigation of success factors for offshore software development outsourcing vendors."Software, IET 6.1 (2012): 1-15.

Appendix A: Guideline Document of the proposed model for SP 1.1

Stage	Action Description	Template(s)
Request	Designated User/Customer **provides** requirements to the Project Team/Business Analyst. • *Only designated users provide requirements. Project Team/Business Analyst should distinguish suitable channels from which to receive requirements.* • *In this step, the provided checklist that was prepared based on specific criteria for distinguishing appropriate requirements providers can be utilized.* • *"User/Customer Profile Template" can also be utilized to help in the process.*	<u>Criteria for distinguishing requirements providers</u> <u>User/customer assessment template</u> <u>User/Customer Profile Template</u>
Understand	Project Team/Business Analyst understands the information contained in the statement of work (SOW), stakeholder profile, and business process flow. • *Project Team/Business Analyst* ***prioritizes*** *the attention of multiple interests.* • *Project Team/Business Analyst* ***categorizes*** *the requirements into different categories, such as functional, technical, performance, and environment.* • *In this stage, the provided checklist can be used to assist in the process.*	<u>Checklist for understanding the requirements</u>
Evaluate	Project Team/Business Analyst **evaluates** the requirements. • *In this step, Project Team/Business Analyst judges the requirements against the objective criteria for the*	<u>Objective</u>

	evaluation of the requirements. The attached objective criteria for the evaluation and acceptance of requirements can be utilized. • *In this step, the "Checklist for Evaluating the Requirements" can be used.*	criteria for the evaluation of requirements Checklist for evaluating the requirements
Accept	Project Team/Business Analyst **accepts** the requirements. • *The attached objective criteria for the acceptance of requirements can be utilized.* • *This step indicates that these requirements are officially accepted and acknowledged for development.* • *In this step, the proposed "Acceptance Form" can be used.*	Objective criteria for the acceptance of requirements Acceptance Form
Finalize	Project Team/Business Analyst and Designated User/Customer finalize the requirements. • *In this step, the requirements are finalized for development by directly involving the designated user/customer.* • *In this step, the agreed set of requirements is produced.*	

Appendix B: Guideline Document of the proposed model for SP 1.2

<table>
<tr><td colspan="3">The relevant stakeholders should "sign off" on the existing requirements (baseline requirements/agreed-upon set of requirements). If a new request is received, the five stages are conducted.</td></tr>
<tr><th>Stage</th><th>Action Description</th><th>Template(s)</th></tr>
<tr><td>Assess</td><td>1. Project Manager/Business Analyst understands the nature of the new requirements or the proposed changes in the requirements for existing commitments.
2. Project Manager/Business Analyst classifies the change as a new requirement or modification of an existing requirement.
3. Project Manager/Business Analyst evaluates/analyzes the impact of this change and performs impact analysis.
4. Project Manager/Business Analyst identifies the impact of the requirement/change.
• In this step, meeting for discussion can be conducted, and the "Minutes of Meeting (MOM) Template" can be used.
• Expert judgment should be employed during the decision-making process.
5. Project Manager/Business Analyst prepares a full impact assessment report.
• In this step, the "Change Request Impact Analysis Form"can be used to help in the process.
• In this step, it is necessary to: (1) include the effort that may be needed for the change, (2) highlight the required cost, project plan, and updated schedules, (3) declare any additional resources necessary for the implementation of the new requirements or the proposed changes.
• If the new requirements or the proposed changes to the existing requirements have no significant impact, the "Negotiate,""Record," and "Commit" stages should take place immediately to enable the stakeholders to sign off on the requirements without delay.</td><td><u>Change Request Impact Analysis Form .doc</u>

<u>Minutes of Meeting.docx</u></td></tr>
</table>

	6. Project Manager/Business Analyst **sends**the impact analysis document with all the relevant details for negotiation. • *In this step, any supporting documents for the impact analysis should be sent.*	
Negotiate	All relevant stakeholders **gather** for a formal discussion (preferable) and provide their views/comments on the change. • *In this step, all relevant stakeholders need to review the new/modified requirement and its impact. The "Review Feedback Document Template" can be used to help in the process.* • *In this step, meeting for discussion can be conducted, and the "Minutes of Meeting (MOM) Template" can be used.*	
Record	Project Manager/Business Analyst **updates** the impact analysis document based on the comments and updates all the documents in the configuration management system. • *In this step, full needed documents are prepared to be submitted to the authority's stakeholders for the "Sign-Off."*	All relevant documents. This includes but is not limited to Change Request Form, Change Request Impact Analysis Form, Change Log, Requirements Specification Document, Test Reports, and all other work products.
Commit	Change Authorization Board **commits** (signs off) on the new/changed requirements and their impact. • *In this step, both the customer and the company acknowledge they have reviewed the proposed change, understood it, and approved it.* • *The signatures provided by them signify they have accepted this proposed change.* • *The "Sign-Off Template" can be used.*	Sign-off Template.doc

Appendix C: Guideline Document of the proposed model for SP 1.3 "manage requirements changes"

Stage	Action Description	Template(s)
Initiate	**1.** Customer (internal/external) **identifies** the need for the change. **2.** Customer (internal/external) **creates** a change request form with a description of the change. **3.** Customer (internal/external) **sends** the change request form to the Project Manager/Business Analyst.	Change Request Form.docx
Validate	**7.** Project Manager/Business Analyst/Project Team **Understands** the nature of the change request ***8.*** Project Manager/Business Analyst/Project Team **Evaluates/Analyzes** the impact of this change and **Performs** Impact Analysis • *This step focuses only on the impact analysis, where the (Project Manager/Business Analyst) analyzes the Change Request Form and understands the impact on the project from various perspectives, such as technical feasibility, timelines, cost, and quality.* • *In this step, the "Requirements Validity Checklist" can be used to determine if a particular requirement is valid.* • *In this step, the "Complexity Checklist" can be used during the analysis to determine if the requirement is complex and the impact it can have on the overall design, cost, schedule, and quality.* **9.** Project Manager/Business Analyst **creates** a detailed Impact Analysis Document. • *In this step, the "Change Request Impact Analysis Form" can be used to help in the process.* • *In this step, the "Complexity Checklist" can be used to help in the process.* **10.** Project Manager/Business Analyst **sends** the Impact Analysis document for validation • *In this step, any supporting documents for the impact analysis, such as the Validity/Complexity Checklist (filled out) should be sent.* **11.** Change Authorization Board **validates** the change.	Requirements Validity Checklist.doc Complexity Checklist (requirements).doc Change Request Impact Analysis Form .doc

	• *In this step, any supporting documents, such as the "Change Request Impact Analysis Form,""Requirements Validity Checklist," and" Complexity Checklist" are used.* • *In this step, meeting for discussion can be conducted and the "Minutes of Meeting (MOM)" Template can be used.* • *In this step, the Change Authorization Board makes the decision," and the Sign-off Template" can be used.* • *The Change Request Log is updated, and the "Change Request Log Template" can be used.*	Minutes of Meeting.docx Sign-off Template.doc Change Request Log.doc
Impleme nt	Project Team **implements** the change in the system. • *In this stage, the change status should be updated in the Change Request Log, and all associated documents should also be updated to reflect the new change.*	
Verify	Project Team **checks** whether the implementation of the recommended change is working correctly. • *In this stage, if the verification phase is not satisfactory, the change request is carried back to the "Validate" stage for the purpose of additional understanding and evaluation.* • *In this stage, the Change Request Log is updated.*	
Update	Project Team **updates** all the documents in the configuration management system.	All relevant documents. This includes but is not limited to the Change Request Form, Change Request Impact Analysis Form, Change Log, Requirements Specification Document, Test Reports, and all other work products.
Release	Project Team deploys the change in the Production/Live environment. The change is released into the production environment, and the final software products are released to the consumer.	

Appendix D: Guideline Document of the proposed model for SP 1.4 “maintaining bidirectional traceability of requirements”

Stage	Action Description	Template(s)
Request	Customer (internal/external) **identifies** the new requirement or modifies the existing requirements.	
Maintain	1. Project Manager/Business Analyst **understands** the request. 2. Project Manager/Business Analyst **analyzes** the new or modified requirement. 3. Project Manager/Business Analyst **identifies** any implied and derived requirements. Project Manager/Business Analyst **updates** software requirements specifications based on the new/modified requirement.	
Validate	1. Project Stakeholders **ensure** all derived requirements are documented and **review** the new/modified requirement and its impact. 2. Project Stakeholders sign-off on the new/modified requirement and its impact.	Review Feedback document.docx Sign-off Template
Allocate	1. Project Manager/Business Analyst updates the requirements traceability matrix. • *In this step, any new/modified requirements are immediately **updated** into the RTM.* 2. Developer **updates** the design specification for the new/changed requirement. • *In this step, the RTM is updated. The design elements are updated against the corresponding requirements in the RTM.* 3. Developer **updates** the code for the new/changed requirement. • *In this step, the RTM is updated. The modified code*	RTM.doc

	references are updated against the corresponding requirements in the RTM.	
Verify	Developer/Tester **updates** the test cases for the new/changed requirement. • *In this step, the RTM is updated. The test case references are updated against the corresponding requirements in the RTM.*	
Release	Developer **updates** the component/product feature and **releases** information for the new/changed requirement. • *In this step, the Project Manager/Business Analyst should ensure that all related documents are updated regularly and revision histories are maintained.*	

Appendix E: Guideline Document of the proposed model for SP 1.5 “ensuring alignment between project work and requirements”

Request	PM/Project team makes a request to review activity at every phase of the development of the software	
Plan	A responsible person ensure a clear understanding of the request PM and Reviewer need to create a review plan that helps conduct the review process of the artifact PM and reviewer need to make sure the plans and work products are both fully ready and available for the review process.	
Review	Technical review of the artifact, RTM, work products, and project plans is undertaken by the reviewer • This stage also involves review of the design, code, and test cases.	Review Feedback document.docx RTM.doc
Identify	The reviewer highlights the findings, list them correctly, and indicate any inconsistencies in the artifact	
Discuss	The reviewer shares the findings with the owner of the work product and the project team, including the project manager. The reviewer and project manager initiate corrective actions in order to resolve these inconsistencies.	Minutes of Meeting.docx
Rework	The reviewer performs the rework process on the work products or project plans based on the initiated corrective actions. • The necessary changes of the work products or project plans are then presented for a verification process to ensure that no feedback is missed. • All associated documents are updated into the system	

Appendix F: Evaluation of the proposed model for each specific practice in REQM process area

Section 1: Cover letter

Dear Survey Participants,

The objective of this project is to implement the ***requirements management (REQM)*** process area of CMMI Level 2, specifically for small- and medium-sized software development organizations. A workflow model for each specific practice in REQM has been developed, and we are seeking qualified individuals to aid in this research study.

If you have experience working with CMMI (Capability Maturity Model Integration), you are invited to participate in this research study. In order to participate, you must complete the attached questionnaire. The following questionnaire will take approximately thirty (30) minutes to complete. Your consent is required in order to participate in this research. Under the confidentiality terms associated with this research, your responses will be kept confidential.

Your participation in this research project will allow for improvements to occur with the implementation of specific practices of REQM. Please note that all data gathered will be used for the purposes of academic research only.

Thank you for your time.

Section 2: Demographics

Evaluator Name: ____________________________________

E-mail Address: ____________________________________

Question 1: Which of these best describes your current position? (Please tick all boxes that apply)

☐ Quality Assurance Manager ☐ Senior Manager

☐ Other (please specify) ______________

Question 2: How long have you worked in Software Engineering/IT? ____years

Question 3: How would you rate your knowledge of CMMI? **____ (**Use a scale of 1 to 5, with **5 = "expert"** and **1 = "little or none**")

Question 4: Have you participated in any previous forms of Process Improvement Assessments, Capability Maturity Assessments, and/or other forms of Process Improvement Appraisals? ______

If yes, please specify (How many?)_____

Section 3: Requirements Management Process Area Overview

3.1 Purpose

The purpose of Requirements Management (REQM) is to manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products.

3.2 Specific Goal and Practice Summary

SG 1 Manage Requirements

- SP 1.1 Understand requirements
- SP 1.2 Obtain commitment to requirements
- SP 1.3 Manage requirements changes
- SP 1.4 Maintain bidirectional traceability of requirements
- SP 1.5 Ensure alignment between project work and requirements

Section 4: Evaluation forms

4.1 Evaluation form of the proposed model for SP 1.1

1. The use of our proposed workflow model for **SP 1.1** would help develop an understanding of "the requirements' providers" with regard to the meaning of the requirements.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

2. How clear is the representation of our proposed workflow model for **SP 1.1**? Using a scale of 1 to 5, with 5 = "very clear" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP 1.1**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for **SP 1.1**? Using a scale of 1 to 5, with 5 = "very useful" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for SP1.1 would improve the software process and lead to the production of high-quality software products.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

6. Our proposed workflow model for **SP 1.1** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

7. Please provide us with comments on how we could improve our proposed workflow model for **SP 1.1.**

4.2 Evaluation form of the proposed model for SP 1.2

1. The use of our proposed workflow model for **SP 1.2** would help ensure that project participants commit to the current and approved requirements as well as the resulting changes in project plans, activities, and work products

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

2. How clear is the representation of our proposed workflow model for **SP 1.2**? Using a scale of 1 to 5, with 5 = "very clear" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

   1.

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP 1.2**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for **SP 1.2**? Using a scale of 1 to 5, with 5 = "very useful" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for **SP 1.2** would improve the software process and lead to the production of high-quality software products.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

6. Our proposed workflow model for **SP 1.2** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

7. Please provide us with comments on how we could improve our proposed workflow model for **SP 1.2.**

4.3 Evaluation form of the proposed model for SP 1.3

1. The use of our proposed workflow model for **SP1.3** would help manage changes to requirements as they evolve during the project.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

2. How clear is the representation of our proposed workflow model for **SP1.3**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP1.3**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for **SP1.3**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for **SP1.3** would improve the software process and lead to the production of high-quality software products.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

6. Our proposed workflow model for **SP1.3** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

7. Please provide us with comments on how we could improve our proposed workflow model for **SP1.3**.

4.4 Evaluation form of the proposed model for SP 1.4

1. The use of our proposed workflow model for **SP1.4** would help maintain the bidirectional traceability of the requirements.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

2. How clear is the representation of our proposed workflow model for **SP 1.4**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP 1.4**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for SP **1.4**? Using a scale of 1 to 5, with 5 = "very" and **1** = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for **SP 1.4** would improve the software process and lead to the production of high-quality software products.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

6. Our proposed workflow model for **SP 1.4** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

7. Please provide us with comments on how we could improve our proposed workflow model for **SP 1.4.**

4.5 Evaluation form of the proposed model for SP 1.5

1. The use of our proposed workflow model for **SP 1.5** would help ensure that project plans and work products remain aligned with requirements. Also, our proposed workflow model for SP1.5 would help find inconsistencies between requirements and project work and, then initiate corrective actions to resolve them.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

2. How clear is the representation of our proposed workflow model for **SP 1.5?** Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP 1.5?** Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for **SP 1.5**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

   ☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for **SP 1.5** would improve the software process and lead to the production of high-quality software products.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

6. Our proposed workflow model for **SP 1.5** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

   ☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

7. Please provide us with comments on how we could improve our proposed workflow model for **SP 1.5.**

Appendix G: Evaluation of Process and Product Quality Assurance (PPQA) process area

Section 1: Cover letter

Dear Survey Participants,

The objective of this project is to implement the ***Process and Product Quality Assurance (PPQA)*** process area of CMMI Level 2, specifically for small- and medium-sized software development organizations. A workflow model for each specific practice in PPQA has been developed, and we are seeking qualified individuals to aid in this research study. If you have experience working with CMMI (Capability Maturity Model Integration), you are invited to participate in this research study. In order to participate, you must complete the attached questionnaire. The following questionnaire will take approximately thirty (30) minutes to complete. Your consent is required in order to participate in this research. Under the confidentiality terms associated with this research, your responses will be kept confidential. Your participation in this research project will allow for improvements to occur with the implementation of specific practices of PPQA. Please note that all data gathered will be used for the purposes of academic research only.

Thank you for your time.

Section 2: Demographics

Evaluator Name: ___________________________________

E-mail Address: ___________________________________

Question 1: Which of these best describes your current position? (Please tick all boxes that apply)

☐ Project Manager **Error! Not a valid embedded object.** ☐ System Engineer

☐ Senior Manager

☐ Other (please specify) _______________

Question 2: How long have you worked in Software Engineering/IT? ____years

Question 3: How would you rate your knowledge of CMMI? ____ **(**Use a scale of 1 to 5, with **5 = "expert"** and **1 = "little or none**")

Question 4: Have you participated in any previous forms of Process Improvement Assessments, Capability Maturity Assessments, and/or other forms of Process Improvement Appraisals? ______

If yes, please specify (How many?)______

Section 3: Process and Product Quality Assurance (PPQA) Process Area Overview

3.1 Purpose

The purpose of Process and Product Quality Assurance (PPQA) is to provide staff and management with objective insight into processes and associated work products.

3.2 Specific Goal and Practice Summary

SG 1 Objectively Evaluate Processes and Work Products

- SP 1.1 Objectively Evaluate Processes
- SP 1.2 Objectively Evaluate Work Products

SG 2 Provide Objective Insight

- SP 2.1 Communicate and Resolve Noncompliance Issues
- SP 2.2 Establish Records

Section 4: Evaluation forms

4.1 Evaluation form of the proposed model for SP 1.1

1. The use of our proposed workflow model for **SP1.1** would help in objectively evaluating the selected performed processes against applicable process descriptions, standards, and procedures.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

2. How clear is the representation of our proposed workflow model for **SP 1.1**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP 1.1**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for **SP 1.1**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for **SP 1.1** would improve the software process and lead to the production of high-quality software products.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

6. Our proposed workflow model for **SP 1.1** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

7. Please provide us with comments on how we could improve our proposed workflow model for **SP 1.1.**

4.2 Evaluation form of the proposed model for SP 1.2

1. The use of our proposed workflow model for SP1.2 would help in objectively evaluating the selected work products against applicable process descriptions, standards, and procedures.

☐ **Strongly Agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly Disagree**

2. How clear is the representation of our proposed workflow model for SP1.2? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for SP1.2? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for SP1.2? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for SP1.2 would improve the software process and lead to the production of high-quality software products.

☐ **Strongly Agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly Disagree**

6. Our proposed workflow model for SP1.2 is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

☐ **Strongly Agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly Disagree**

7. Please provide us with comments on how we could improve our proposed workflow model for SP 1.2.

4.3 Evaluation form of the proposed model for SP 2.1

1. The use of our proposed workflow model for **SP 2.1** would help in communicating the quality issues and ensuring the resolution of noncompliance issues with the staff and managers.

☐ **Strongly Agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly Disagree**

2. How clear is the representation of our proposed workflow model for **SP 2.1**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP 2.1**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

4. How useful would it be to the software industry to use our proposed workflow model for **SP 2.1**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ 5 ☐ 4 ☐ 3 ☐ 2 ☐ 1

5. The use of our proposed workflow model for **SP 2.1** would improve the software process and lead to the production of high-quality software products.

☐ **Strongly Agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly Disagree**

6. Our proposed workflow model for **SP 2.1** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

☐ **Strongly agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly disagree**

7. Please provide us with comments on how we could improve our proposed workflow model for **SP 2.1.**

4.4 Evaluation form of the proposed model for SP 2.2

1. The use of our proposed workflow model for SP 2.2 would help in establishing and maintaining the records of quality assurance activities.

☐ **Strongly agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly disagree**

2. How clear is the representation of our proposed workflow model for **SP 2.2**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ **5** ☐ **4** ☐ **3** ☐ **2** ☐ **1**

3. How much knowledge of CMMI is required to learn how to use our proposed workflow model for **SP 2.2**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ **5** ☐ **4** ☐ **3** ☐ **2** ☐ **1**

4. How useful would it be to the software industry to use our proposed workflow model for **SP 2.2**? Using a scale of 1 to 5, with 5 = "very" and 1 = "not at all":

☐ **5** ☐ **4** ☐ **3** ☐ **2** ☐ **1**

5. The use of our proposed workflow model for **SP 2.2** would improve the software process and lead to the production of high-quality software products.

☐ **Strongly agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly disagree**

6. Our proposed workflow model for **SP 2.2** is applicable to small- and medium-sized software development organizations. In other words, it can be applied to both small- and medium-sized software development organizations.

☐ **Strongly agree** ☐ **Agree** ☐ **Neutral** ☐ **Disagree** ☐ **Strongly disagree**

7. Please provide us with comments on how we could improve our proposed workflow model for **SP 2.2**.

Appendix H: Checklists and templates for each specific practice in REQM process area

Section 1: SP 1.1 – "Understand requirements"

1.1 Criteria for distinguishing appropriate requirements providers

Project Name	
Project ID	
Prepared by	
Date:	

No.	Criteria	Question
1	Necessity of involvement	If this user/customer was not included, engaged, and involved during the development process, can he/she derail the process?
2	Contribution	Does this user/customer have useful information about the organization? Does this user/customer have knowledge about the product? Does this user/customer have the enough knowledge to contribute to the process?
3	Influence	Does this user/customer have decision making authority? Does this user/customer have strong power to manage decisions during the development process? Does this user/customer have the ability to assess impact of a change in the product?
4	Technical know	Does this user/customer know some technical issues about the product?
5	Level of commitment	Does this user/customer have strong level of commitment to the process and the product?

1.2 Customer assessment template

Project Name				
Project ID				
Prepared by				
Date:				
Stakeholder ID#________________________________				
Question	Type of Assessment			
	High	Medium	Low	
If this user/customer was not included, engaged and involved during the development process, can he/she derail the process?				
Does this user/customer have the enough knowledge to contribute to the process?				
Does this user/customer have knowledge about the product?				
Does this user/customer have useful information about the organization?				
Does this user/customer have decision making authority?				
Does this user/customer have the ability to assess impact of a change in the product?				
Does this user/customer have strong power to manage decisions during the development process?				
Does this user/customer know some technical issues about the product?				
Does this user/customer have strong level of commitment to the process and the product?				

1.3 Customer profile template

Project Name	
Project ID	
Prepared by	
Date: dd/mm/yyyy	

Name	Role	Contact information	The way of communication	Project Motivation & Interests	Project Impact	Status

1.4 Checklist for understanding the requirements

Project Name					
Project ID					
Prepared by					
Date:					
No.	Item to check	**Yes**	**NO**	**N/A**	**Remarks**
1	Are user/customer correctly designated and identified to provide and prove the requirements?				
2	Are the requirements provided by designated user/customer?				
3	Is SOW sufficiently understood to initiate and begin the project?				
4	Does PM understand each requirement provided by designated user/customer?				
5	Are all types of requirements (such as functional, technical, performance, and environment) provided by designated user/customer?				
6	Are all types of requirements correctly categorized into different categories such as functional, technical, performance and environment?				
7	Is the business process flow understood for all types of requirements?				
8	Are the inputs and outputs distinguished in the business process flow for all types of requirements?				

1.5 Objective criteria for the evaluation of requirements

Project Name		
Project ID		
Prepared by		
Date:		
NO	Criteria	Question
1	Clarity	Is each requirement properly stated? Does each requirement have only one interpretation?
2	Feasible	Can each requirement be implemented within available resources or technology?
3	Consistent	Is each requirement consistent within itself and with one another in the document?
4	Unique	Is each requirement uniquely defined with no duplication?
5	Prioritized	Is each requirement defined business for development?
6	Achievable	Can each requirement be achieved according to the budget and time constraints?

1.6 Checklists for evaluating the requirements

No.	Item to check	Yes	NO	N/A	Remarks
1	Is each requirement properly stated?				
2	Does each requirement have only one interpretation?				
3	Can each requirement be implemented within available resources or technology?				
4	Is each requirement consistent within itself and with one another in the document?				
5	Is each requirement uniquely defined with no duplication?				
6	Can each requirement be achieved according to the budget and time constraints?				
7	Does each requirement have only one interpretation?				
8	Is each requirement traced to the main source?				
9	Do the documents follow the standards?				

1.7 Objective criteria for the acceptance of requirements

NO.	Criteria	Question
1	Clarity	Is each requirement properly stated? Does each requirement have only one interpretation?
2	Feasible	Can each requirement be implemented within available resources or technology?
3	Consistent	Is each requirement consistent within itself and with one another in the document?
4	Unique	Is each requirement uniquely defined with no duplication?
5	Prioritized	Is each requirement defined business for development?
6	Achievable	Can each requirement be achieved according to the budget and time constraints?

1.8 Acceptance form

Project Name	
Project ID	
Prepared by	
Date: dd/mm/yyyy	

<table>
<tr><td>

This is to indicate that the requirements, (that met all the objective criteria for the evaluation), are officially accepted and acknowledged for development.

***(Insert comments, if required)**
Signature _________________________________

</td></tr>
</table>

Section 2: SP1.2–"Obtain commitment to requirements"

2.1 Change Request Impact Analysis Form

DETAILS	
Request ID:	
Project Name :	
Project ID:	
Change Initiator:	
Submitted To:	
Date:	
DESCRIPTION OF CHANGE	
-- -- --	
REQUESTOR PRIORITY	
--	
TOTAL EFFORT "IN PERSON DAYS"	
--	
START DATE: dd/mm/yyyy	
--	
END DATE: dd/mm/yyyy	
--	
TECHNICAL IMPACT	
-- -- -- --	

RISKS/ISSUES
--   --
COMMERCIAL IMPACT
---   ---
OTHER IMPACT
---   ---
***(Insert comments, if required)**

Authorized By

Company Name		**Customer Name**	
Role/Title:		**Role/Title**	
Date		**Date**	
Sign-Off		**Sign-Off**	
***(Insert comments, if required)**		***(Insert comments, if required)**	

2.2 Minutes of Meeting Template

Request ID:

Date: dd/mm/yyyy

Project Name :

Project ID:

Prepared By:

Submitted To:

- Meeting Location
  1. Building:______________________
  2. Conference Room:______________
  3. Conference Line:________________

- Meeting Date: dd/mm/yyyy_________________
- Meeting Start <HH:MM> ___________________

- Attendees:
  1. ___________________________
  2. ___________________________
  3. ___________________________
  4. ___________________________
  5. ___________________________

- Meeting's Action Items

Action	Assigned To	Deadline
<Action Item>	<Assignee>	<mm/dd/yy>

- Decisions Made
  1. Decision 1:______________________
  2. Decision 2:______________________
  3. Decision 3:______________________

- Next Meeting Date & Time (If required)

2.3 Sign-Off Template

Request ID:
Date: dd/mm/yyyy
Project Name :
Project ID:
Prepared By:
Submitted To:

Both <customer Name> and <Company name> acknowledge that they have reviewed the proposed change, understood it.

The signature below signifies that <customer Name> and <Company name> have accepted/rejected/batched this proposed change.

Company Name		Customer Name	
Role/Title:		Role/Title	
Date		Date	
Sign-Off		Sign-Off	
*(Insert comments, if required)		*(Insert comments, if required)	

Section 3: SP 1.3 – "manage requirements changes"

3.1 Change Request Impact Analysis Form

DETAILS	
Request ID:	
Project Name :	
Project ID:	
Change Initiator:	
Submitted To:	
Date: dd/mm/yyyy	
DESCRIPTION OF CHANGE	
--	
REQUESTOR PRIORITY	
--	
TOTAL EFFORT "IN PERSON DAYS"	
--	
START DATE: dd/mm/yyyy	
--	
END DATE: dd/mm/yyyy	
--	
TECHNICAL IMPACT	
--	
RISKS/ISSUES	
--	
--	
COMMERCIAL IMPACT	
--	
--	

OTHER IMPACT

***(Insert comments, if required)**

Authorized By

Company Name		Customer Name	
Role/Title:		Role/Title	
Date		Date	
Sign-Off		Sign-Off	
*(Insert comments, if required)		*(Insert comments, if required)	

3.2 Minutes of Meeting Template

Request ID:
Date: dd/mm/yyyy
Project Name :
Project ID:
Prepared By:
Submitted To:

- Meeting Location
  4. Building:______________________
  5. Conference Room:______________
  6. Conference Line:________________

- Meeting Date: dd/mm/yyyy_________________
- Meeting Start <HH:MM> ___________________

- Attendees:
  6. __________________________
  7. __________________________
  8. __________________________
  9. __________________________
  10. __________________________

- Meeting's Action Items

Action	Assigned To	Deadline
<Action Item>	<Assignee>	<mm/dd/yy>

- Decisions Made
  4. Decision 1:______________________
  5. Decision 2:______________________
  6. Decision 3:______________________

- Next Meeting Date & Time (If required)

3.3 Sign-Off Template

Request ID:
Date: dd/mm/yyyy
Project Name :
Project ID:
Prepared By:
Submitted To:
Both <customer Name> and <Company name> acknowledge that they have reviewed the proposed change, understood it. The signature below signifies that <customer Name> and <Company name> have accepted/rejected/batched this proposed change.

Company Name		Customer Name	
Role/Title:		Role/Title	
Date		Date	
Sign-Off		Sign-Off	
*(Insert comments, if required)		*(Insert comments, if required)	

3.4 CHANGE REQUEST FORM

DETAILS	
Request ID:	
Project Name :	
Project ID:	
Change Initiator:	
Submitted To:	
Date: dd/mm/yyyy	

DESCRIPTION OF CHANGE

IMPACT OF CHANGE
How the system behaved before & after implementing the change
----

PLAN

***(Insert comments, if required)**

APPROVAL STATUS

***(Insert comments, if required)**

Name:______________
Signature ___________ **Date**

REVISION HISTORY

Version	Date	Name	comments	Signat ure

3.5 Requirements Validity Checklist

Criteria	Question	Response
Clarity	Is each requirement properly stated? Does each requirement have only one interpretation?	
Feasible	Can each requirement be implemented within available resources or technology?	
Consistent	Is each requirement consistent within itself and with one another in the document?	
Unique	Is each requirement uniquely defined with no duplication?	
Prioritized	Is each requirement defined business for development?	
Achievable	Can each requirement be achieved according to the budget and time constraints?	
Appropriateness	Are the requirements appropriate to implement?	
Completeness	Does the requirement look complete?	

3.6 Complexity Checklist (requirements)

Area	Question	Response
Architecture	Will the product architecture be impacted?	
	Is it a new functional area?	
	Is it a fundamental change to the current area?	
Novelty	Do we need to use new technology?	
	Is it new? We're going to do something that we never did before.	
Complexity	Does it require lots of time?	
	Does it require lots of effort?	

3.7 Change Log Sheet

Request ID	Date:	Change Description	Initiated by	Effort Estimate	Cost Estimate	Impact	Approved by	Status	comments

Section 4: SP 1.4–"maintaining bidirectional traceability of requirements"

4.1 Document Review

Document Details
Date: dd/mm/yyyy
Project Name :
Project ID:
Document Type:
Author:
Reviewer Name:
Submitted To:

REVISION HISTORY

Version	Date	Author	comments	Signature

COMMENTS

No.	Location	Severity High/ Medium/Low	Type	Description	Cleared (Yes/No)
1					
2					
3					
4					

* Type such as (Error , Omission, Inconsistency, Unclassified, Query, Comment only, Standards non-compliance, Typographical, grammatical or spelling)

4.2 Requirement Traceability Matrix

DETAILS	
Request ID:	
Project Name :	
Project ID:	
Prepared by:	
Submitted To:	
Date: dd/mm/yyyy	
Version #	

	REQUIREMENTS DETAILS						Design Reference/Code Reference/		Test Case Reference			Traced to Completion / Comments	
No	Req. ID	Req. Name	Associated ID(s)	Customer Need(s)	Description	Functional Requirement	Design Document Ref.	Code Ref.	Functional Test Case	Unit Test Case	Integration Test case	Traced to Completion	Comments
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

4.3 Sign-Off Template

Request ID:
Date: dd/mm/yyyy
Project Name :
Project ID:
Prepared By:
Submitted To:
Both <customer Name> and <Company name> acknowledge that they have reviewed the proposed change, understood it. The signature below signifies that <customer Name> and <Company name> have________________________________ This proposed change.

Company Name		Customer Name	
Role/Title:		Role/Title	
Date		Date	
Sign-Off		Sign-Off	
*(Insert comments, if required)		*(Insert comments, if required)	

Appendix I: Proposed Audit checklist for SP 1.1 – "Objectively evaluate processes"

Table 24: Proposed Audit checklist for SP 1.1 – "Objectively evaluate <u>processes</u>"

Project Information		
Project Name		
Project Type		
Date of Process Audit		
Project Manager		
Process Auditor		
Evaluate each item in the audit checklist against the criteria of SCAMPI Class A appraisal: "FI" – Fully Implemented "LI" – Largely Implemented "PI" – Partially Implemented "NI" – Not Implemented "NY"– Not yet Implemented		
Process Area	specific practice	Item to check
Requirements Management (REQM)	SP 1.1	Does the software project team develop an understanding of "the requirements' providers" with regard to the meaning of the requirements?
		Are the requirements evaluated to make sure that objective criteria are met?
	SP 1.2	Are the commitments negotiated and recorded?
		Is the impact of requirements assessed on existing commitments?
	SP 1.3	Are all requirements changes recorded and documented?
		Is change data made available and accessible to the project?
	SP 1.4	Is requirements traceability maintained?
		Is RTM generated?
	SP 1.5	Are the software project plans and work products consistent with both requirements and the changes made to them?
		Are necessary corrective actions initiated?
Process and Product Quality Assurance (PPQA)	SP 1.1	Are the evaluation criteria established and maintained? Are they used to evaluate selected processes?
		Is each noncompliance found during the evaluation clearly identified?
	SP 1.2	Are the evaluation criteria established and maintain? Are they

		used to evaluate selected work products?
		Is each noncompliance found during evaluations clearly identified?
	SP 2.1	Is each noncompliance resolved? Are noncompliance issues tracked to resolution?
		Are non-compliance issues and obtained results communicated to all stakeholders?
	SP 2.2	Are product quality assurance tasks clearly recorded?
		Is the status of quality assurance activities revised when necessary? Are observations and findings well-recorded in the audit report?
Project Planning (PP)	SP 1.1	Is the work breakdown structure (WBS) developed?
		Are the work packages defined in enough detail?
	SP 1.2	Are the attributes of work products estimated?
		Is the technical approach for the software project determined?
	SP 1.3	Are life cycle phases for each project established?
	SP 1.4	Are the efforts and cost for work products and tasks estimated using historical data and models?
		Are supporting infrastructure needs included and utilized to estimate effort and cost?
	SP 2.1	Are the main milestones, constraints, and schedule assumptions identified?
		Is the budget for the software project established?
	SP 2.2	Are project risks identified, documented, and revised as appropriate?
	SP 2.3	Are privacy and the security of data maintained? Are the requirements for providing access to project data determined?
		Is there a procedure to archive project data? Is there a procedure to access project data which were archived?
	SP 2.4	Are the requirements (process, communication and staffing) determined?
		Are the facility and component requirements determined?
	SP 2.5	Are the skills and knowledge required to conduct a software project identified? Are they assessed?
		Are there mechanisms for providing required the skills and knowledge?
	SP 2.6	Is the stakeholder involvement plan developed?
	SP 2.7	Is the overall project plan developed?
	SP 3.1	Are interrelated plans reviewed?
	SP 3.2	Are the methods and estimating parameters revised when necessary?
		Are requirements and schedules revised, if needed?

	SP 3.3	Are internal/ external commitments reviewed with the appropriate senior management level?
		Are commitments negotiated with concerned stakeholders?
Configuration Management (CM)	SP 1.1	Are configuration items selected based on provided criteria?
		Is each configuration item uniquely defined with no duplication? Are the significant features of each configuration item specified?
	SP1.2	Is a mechanism to monitor and direct multiple levels of control established and identified?
		Are configuration items stored in a configuration management system? Can they be retrieved from the configuration management system?
	SP1.3	Before Creating or releasing baselines, Is authorization obtained from the CCB?
		Is the set of configuration items contained in a baseline documented?
	SP 2.1	Is the level of change request traced to closure?
		Is the impact of changes analyzed? Are changes categorized and prioritized?
	SP2.2	Are changes to configuration items controlled throughout the life of the product or service?
	SP 3.1	Are configuration management actions recorded in detail?
		Is the latest version of baselines specified?
	SP 3.2	Is the integrity of baselines assessed?
		Is a configuration management audit conducted? Are action items tracked from the audit to closure?
Measurement and Analysis (MA)	SP 1.1	Are information needs established and documented? Are they prioritized?
		Is traceability of measurement objectives established and maintained?
	SP 1.2	Are candidate measures identified based on the recorded measurement objectives?
		Are measures prioritized, reviewed, and updated?
	SP1.3	Are the existing sources of data identified? Are data collection mechanisms well-established?
		Is data collection prioritized and examined?
	SP1.4	Are appropriate data analysis mechanisms selected?
		Are the analyses to be conducted specified and prioritized? Are the criteria for evaluating the analysis results specified?
	SP 2.1	Is data obtained for base measures? Is data generated for derived measures?

		Are data integrity checks provided?
	SP2.2	Are initial analyses conducted? Are conclusions drawn?
		Are the results reviewed with relevant stakeholders
	SP2.3	Is the collected data examined and reviewed in terms of completeness, accuracy, and integrity?
		Is data stored according to data storage procedure?
	SP2.4	Check if relevant stakeholders assisted in understanding results
Supplier Agreement Management (SAM)	SP1.1	Is the type of acquisition determined?
	SP1.2	Are the criteria for evaluating potential suppliers Established?
		Are the proposals, risks of each proposed supplier, and proposed suppliers' abilities evaluated?
	SP1.3	Is the supplier agreement documented? Is it periodically examined?
		Is the supplier agreement revised when necessary? Are the project's plans and commitments revised when necessary?
	SP2.1	Is supplier progress monitored?
		Are work products selected and evaluated from the supplier?
	SP2.2	Are the acceptance procedures defined? Are the results of the acceptance test documented?
		Is action plan established when necessary?
	SP2.3	Check if acquired products are stored, integrated, and distributed as appropriate
Project Monitoring and Control (PMC	SP 1.1	Is the progress monitored and examined against the schedule?
		Are (project's costs, the attributes of work products and tasks, resources used, and the knowledge and skills of project staff) monitored?
	SP2.1	Are external and internal commitments regularly reviewed?
		Are obtained results of commitment reviews documented?
	SP 1.3	Is the documentation of risks regularly reviewed?
		Is the documentation of risks revised, when necessary? Is the risk status communicated to relevant stakeholders?
	SP1.4	Are data management tasks examined as stated in the software project plan?
		Are the significant issues identified and documented? Are their impacts identified and documented?
	SP1.5	Is the level of stakeholder involvement periodically monitored and reviewed?
		Are the significant issues identified and documented? Are their impacts identified and documented?
	SP1.6	Is the status of activities and work products regularly

		communicated to relevant stakeholders?
		Are the deviations from the plan Identified and documented?
	SP1.7	Are milestone reviews conducted with relevant stakeholders?
		Are the significant issues identified and documented? Are their impacts identified and documented?
	SP 2.1	Are gathered issues analyzed?
	SP 2.2	Are the appropriate actions required to tackle identified issues determined and documented?
		Are changes negotiated?
	SP 2.3	Are corrective actions monitored?
		Are obtained results of corrective actions analyzed?

Appendix J: Proposed Audit checklist for SP 1.2 – "Objectively evaluate work products"

Table 25: Proposed Audit checklist for SP 1.2 – "Objectively evaluate work products"

Project Information		
Project Name		
Project Type		
Date of Process Audit		
Project Manager		
Process Auditor		
Evaluate each item in the audit checklist against the criteria of SCAMPI Class A appraisal: "FI" – Fully Implemented "LI" – Largely Implemented "PI" – Partially Implemented "NI" – Not Implemented "NY"– Not yet Implemented		
No.	**Item to check**	
1	**Generic** Check Points	Check if the latest version of the document/template is used as stated in the plan
2		Check if all sections of the document/template are properly filled.
3		Check if all sections of the document/template are filled out in clear and concise language.
4		Check if document/template name is correct.
5		Check if the purpose/scope of the document/template is captured.
6		Check if all the reference documents are identified
7		Check if all abbreviations included in document/template are defined
8		Check if the introduction section of document/template provides overview information about it.
9		Check if work product is included under configuration management control as stated in the plan
10		Check if the work product history is updated.
11	Check Points Related to **design**	Check if alternative design solutions are identified
		Check if selection criteria for selecting a solution are available
12		Check if all the identified design modules are described in sufficient detail
13		Check if dependencies with other design elements are identified and documented.
14		Check if interfaces of design are defined, and their operations are

		described in sufficient detail.
15		Check if requirements associated with design are provided.
16		Check if all the reference documents associated with design are identified
17		Check if traceability matrix is updated
18	Check Points related to **code**	Check if the code satisfies the coding standards as stated in the software project plan
19		Check if the code review comments are correctly closed
20		Check if the traceability matrix is updated
21	Check Points related to **Requirements Document**	Check if functional requirements are identified and documented in clear and concise language.
22		Check if interface requirements are identified and documented in clear and concise language.
23		Check if performance requirements are identified and documented in clear and concise language.
24		Check if the traceability matrix is updated
25	Check Points related to **Test Cases Document**	Check if all the steps to conduct the test are identified and documented.
26		Check if all the pre-requisites needed to execute the test are identified and documented.
27		Check if the testing is conducted and the obtained results are available
28		Check if the traceability matrix is updated

Appendix K: Guideline Document of the proposed model for SP 1.1

stage	Action Description	Template(s)

Plan	• SQA Manager creates an audit plan.
	  ○ *The created audit plan should clearly describe which process/work product is to be audited, for what purpose, at what frequency, and by whom.*
	  ○ *Note: audit plan can be (yearly / bi annual / quarterly)*
	• SQA Manager identifies independent PPQA auditor.
	• SQA Manager finalizes Project audit schedule with the Project Manager
		Criteria to select a key process to be audited
		
		Criteria to select auditor/reviewer

Prepare	• Project Manager assists PPQA auditor in obtaining the required information/ artifacts • PPQA auditor identifies the required *audit checklist for the process* to be audited • PPQA auditor reviews associated Plans, Project documents and any other project artifacts	audit checklist for the process to be audited
Audit	• PPQA auditor conducts the process audit based on defined criteria and audit instructions • PPQA auditor identifies noncompliance issues, improvements, and best practices • PPQA auditor analyzes the root cause and impact of non-compliance. • PPQA auditor discusses observations, associated lessons learnt and corrective actions with relevant process owners.	
Report	• PPQA auditor records observations in an audit report. • PPQA auditor places audit document in the project records. • The final version of the audit report might need to be distributed to concerned stakeholders and users	

Appendix L: Guideline Document of the proposed model for SP 1.2

stage	Action Description	Template(s)

Plan	• SQA Manager creates an audit plan.
	  ○ *The created audit plan should clearly describe which process/work product is to be audited, for what purpose, at what frequency, and by whom.*
	  ○ *Note: audit plan can be (yearly / bi annual / quarterly)*
	• SQA Manager Identifies independent PPQA auditor.
	• SQA Manager Finalize Project audit schedule with the Project Manager
		Criteria to select a key work product to be audited.
		
		Criteria to select auditor/reviewer

Prepare	• Project Manager assists PPQA auditor in obtaining the required information/ artifacts • PPQA auditor identifies the required *audit checklist for the work product* to be audited • PPQA auditor reviews associated Plans, Project documents and any other project artifacts	audit checklist for some important work products
Audit	• PPQA auditor conducts the work product audit based on defined criteria and audit instructions • PPQA auditor identifies noncompliance issues, improvements, and best practices • PPQA auditor analyzes the root cause and impact of non-compliance. • PPQA auditor discusses observations, associated lessons learnt and corrective actions with relevant owners.	
Report	• PPQA auditor records observations in an audit report. • PPQA auditor places audit document in the project records.	

	• The final version of the audit report might need to be distributed to concerned stakeholders and users	

Appendix M: Guideline Document of the proposed model for SP 2.1

stage	Action Description	Template(s)
Resolve	• SQA Manager reviews Audit report and publish to stakeholders for action • SQA Manager and project Manager work with the team to ensure that the issues highlighted in the audit report are addressed in a timely manner • In cases where there is no feasible resolution to noncompliance issues, the escalation path will be taken.	Audit report will include audit checklist for the process to be audited and audit checklist for some important work products
Escalate	• SQA Manager Analyses root cause for the issue, identify solution, facilitate necessary steps for resolution • SQA Manager Identifies if the resolution is feasible • If the issue still cannot be resolved, it is escalated and reported by SQA Manager to the appropriate level of authority, as defined in the project management plan.	
Follow-up	• Operative Process Owners and Project SQA follow-up with team members periodically to verify resolution. • Operative Process Owners and Project SQA ensure smooth closure of the issues.	
Find-out	• Project SQA records observations in an audit report. • Project SQA analyzes in depth the non compliances in the project across processes and work products. • Project SQA Observes/Records similarities and trends. • Project SQA addresses specific issues with process or	

	resources to ensure effective implementation and adoption of project process. • Project SQA compares the analysis results obtained with other projects, as well as the expectations they may have as a result of previous experiences, so as to bring about process improvement at organizational level.	

Appendix N: Guideline Document of the proposed model for SP 2.2

Stage	Action Description	Template(s)
Record	• SQA Manager records and documents the status of quality assurance activities • SQA Manager Consolidates the project level PPQA records to create organizational level PPQA records.	Audit Records will include audit checklist for the process to be audited and audit checklist for some important work products
Revise	• SQA Manager ensures that the PPQA records established in the previous stage, "record," are kept up to date. • If the status of PPQA activities is not found to be recent, then SQA Manager revises the PPQA records with the latest information so that they are always kept up to date.	
Share	• SQA Manager shares the quality report at predefined intervals with the managers and process engineering group	
Improve	• SQA manager examines the quality report and other discussion forums that are used to identify issues, gaps, best practices, and trends as part of process improvement. ○ Note: All obtained suggestions and observations are recorded in detail and presented to management for approval. Based on the obtained approval, the process improvement	

	initiatives and changes are institutionalized in an organization-wide manner. • SQA manager carries out the process improvement at the organizational level.	

APPENDIX O

SPI expert	Job title	Experience of SPI expert (years)	Knowledge of CMMI (Low 1 - 5 High)	Qualification	More information	Organization	Brief Background about organization
1	Project manager	15	4	Masters in Management (Project Management)	Project Management Specialist (Project Management Office) PMO PMP certified Scrum Fundamentals Certified Participated in more than 15 software process assessments	Org. 1	This organization consists of the following departments: • The Computing Services and Support (CSS) department serves the organization staff with extensive computing services • The Administrative Information Systems (ADIS) department maintains the administrative applications • The Project Management Office (PMO) department supports information technology projects by following and facilitating the use of processes such as initiating, planning, executing, controlling and closing IT projects. • The Business Support Department (BSD) department plays an important support role by facilitating all business-related and administrative tasks. • Systems Operations and Support (SOS) department provides systems and operational support to different operating system platforms.
2	Project manager	13	4	Master of Science in Software Engineering	PMP certified Participated in 15 software process assessments	Org. 2	• Software development organization • It builds software solutions for the financial services, hospitality, • Fixing some of the industrial application. .
3	Project manager	9	4	Master of Science in Software Engineering	PMP certified Participated in 10 software process assessments	Org. 1	This organization consists of the following departments: • The Computing Services and Support (CSS) department serves the organization staff with extensive computing services • The Administrative Information Systems (ADIS) department maintains the administrative applications • The Project Management Office (PMO) department supports information technology projects by following and facilitating the use of processes such as initiating, planning, executing, controlling and closing IT projects. • The Business Support Department (BSD) department plays an important support role by facilitating all business-related and administrative tasks. • Systems Operations and Support (SOS) department provides systems and operational support to different operating system platforms.

4	Software developer	7	3	Master of Science in Software Engineering	Participated in 6 software process assessments	Org. 1	This organization consists of the following departments: • The Computing Services and Support (CSS) department serves the organization staff with extensive computing services • The Administrative Information Systems (ADIS) department maintains the administrative applications • The Project Management Office (PMO) department supports information technology projects by following and facilitating the use of processes such as initiating, planning, executing, controlling and closing IT projects. • The Business Support Department (BSD) department plays an important support role by facilitating all business-related and administrative tasks. • Systems Operations and Support (SOS) department provides systems and operational support to different operating system platforms.
5	Software developer	10	4	Master of Science in Software Engineering	PMP certified Participated in 5 software process assessments	Org. 3	• Software development & design organization • It has web designers, programmers • It provides web solutions in web content planning, designing, implementation and management • It uses PDIM methodology (Plan, Design, Implement, and Manage)

Table 26: Brief Background about the SPI experts and their organizations

SPI expert	Job title	Experience of SPI expert (years)	Knowledge of CMMI (Low 1 - 5 High)	Qualification	More information	Organization	Brief Background about organization
1	Quality Officer/IT Auditor	10	4	Master of Science in Software Engineering	ISO 20000 Lead Auditor	Org. 2	• Software development organization • It builds software solutions for (financial services and hospitality) • Fixing some of the industrial application. .
2	Quality Officer/IT Auditor	12	4	Master of Science in Software Engineering	Certified Information Systems Auditor® (CISA)	Org. 3	• Software development & design organization • It has web designers, programmers • It provides absolute web solutions in web content planning, designing, implementation and management • It uses PDIM methodology (Plan, Design, Implement, and Manage)
3	Project manager	15	4	Masters in Management (Project Management)	Project Management Specialist (Project Management Office) PMO PMP certified Scrum Fundamentals Certified Participated in more than 15 software process assessments	Org. 1	This organization consists of the following departments: • The Computing Services and Support (CSS) department serves the organization staff with extensive computing services • The Administrative Information Systems (ADIS) department maintains the administrative applications • The Project Management Office (PMO) department supports information technology projects by following and facilitating the use of processes such as initiating, planning, executing, controlling and closing IT projects. • The Business Support Department (BSD) department plays an important support role by facilitating all business-related and administrative tasks. • Systems Operations and Support (SOS) department provides systems and operational support to different operating system platforms.

4	Project manager	13	4	Master of Science in Software Engineering	PMP certified Participated in 10 software process assessments	Org. 1	This organization consists of the following departments: • The Computing Services and Support (CSS) department serves the organization staff with extensive computing services • The Administrative Information Systems (ADIS) department maintains the administrative applications • The Project Management Office (PMO) department supports information technology projects by following and facilitating the use of processes such as initiating, planning, executing, controlling and closing IT projects. • The Business Support Department (BSD) department plays an important support role by facilitating all business-related and administrative tasks. • Systems Operations and Support (SOS) department provides systems and operational support to different operating system platforms.
5	Software developer	7	3	Master of Science in Software Engineering	Participated in 6 software process assessments	Org. 1	This organization consists of the following departments: • The Computing Services and Support (CSS) department serves the organization staff with extensive computing services • The Administrative Information Systems (ADIS) department maintains the administrative applications • The Project Management Office (PMO) department supports information technology projects by following and facilitating the use of processes such as initiating, planning, executing, controlling and closing IT projects. • The Business Support Department (BSD) department plays an important support role by facilitating all business-related and administrative tasks. • Systems Operations and Support (SOS) department provides systems and operational support to different operating system platforms.
6	Software developer	8	4	Masters in Management (Project Management)	Certified Systems Auditor	Org. 3	• Software development & design organization • It has web designers, programmers • It provides absolute web solutions in web content planning, designing, implementation and management

Table 27: Brief Background about the SPI experts and their organizations

Table 28 Criteria to select the auditor/reviewer

No.	Question	Scale
1	Does this person have good documentation skills?	0: Not involved in any previous software process assessments in the current organization 0.5: Involved in (1-5) previous software process assessments in the current organization 1: involved in (+ 10) previous software process assessments in the current organization
2	Does this person have good verbal skills?	0: Not involved in any previous software process assessments in the current organization 0.5: Involved in (1-5) previous software process assessments in the current organization 1: involved in (+10) previous software process assessments in the current organization
3	Dose this person have enough knowledge to contribute to the process/work product?	0: did not Fully participating in auditing for any artifacts from previous similar projects 0.5: Fully participating in auditing (1-15) artifacts from previous similar projects 1: Fully participating in auditing (+16) artifacts from previous similar projects
4	Does this person have the ability to assess the impact of non-compliances?	0: did not Fully participating in auditing any artifacts from previous similar projects 0.5: Fully participating in auditing (1-15) artifacts from previous similar projects 1: Fully participating in auditing (+16) artifacts from previous similar projects
5	Does this person have the appropriate training to conduct auditing activities?	0: Did not attend any approved training course on “auditing activities” 0.5: attended (1-3) approved training course on “auditing activities” 1: attended (+6) approved training course on “auditing activities”
6	Does this person have strong enough power to bring both problems and non-compliances to light?	0:Did not officially certified as auditor by any institution/organization 1:Officially certified as auditor by any institution/organization

VITAE

Personal Details

Name:	Ismail Mohamed Hemdan Keshta
Date of Birth:	October 20 , 1985
Place of Birth:	Saudi Arabia
Nationality:	Palestinian
Marital Status:	Single
E-mail:	ismailk@kfupm.edu.sa g200376970@kfupm.edu.sa s237697@hotmail.com
Mobile:	+966506229687
Current address :	P.O. Box 7802, Dhahran 31261, Saudi Arabia
Permanent address :	AlQzaan Street, Al Riyadh, Saudi Arabia

Education

King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia	**Ph.D. in Computer Sciences and Engineering (double major), 2016, GPA 3.75 out of 4.0.** Dissertation Title: "Towards achieving CMMI Level 2 for small and medium sized software development organizations."
King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia	**Master of Science degree in Computer Engineering, June 2011, GPA 3.714 out of 4.0** Thesis Title: "Performance Analysis of Real-Time Publish/Subscribe Middleware For Wireless Sensor Networks."
King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia	**Bachelor of Sciences degree in Computer Engineering, February 2009**, Cumulative **GPA 3.66 out of 4.0 and Major GPA 3.712 out of 4.0.**

Academic Work Experience

Position	Year	University	Course Code	Course Title
Lecturer	2011-2012	**Imam** Muhammad ibn Saud Islamic University, Riyadh, Saudi Arabia	CS 052	Computer Skills For Preparatory Programs “Applied Track” (61 students)
			CS 029	Computer Skills For Preparatory Programs "languages and translation Track” (17 students)
Lecturer B (LB)	2012-2016	King Fahd University of Petroleum and Minerals (**KFUPM**), Dhahran, Saudi Arabia	COE301/ ICS233	Computer Organization /Computer Architecture & Assembly Language (35 students)
			ICS 103	Computer Programming in C (49 students)
			PYP 002	Preparatory Computer Science (25 students)

Honors and Awards

- Ph.D. scholarship award, KFUPM, Dhahran, Saudi Arabia (2012-2016)
- MSc scholarships award, KFUPM, Dhahran, Saudi Arabia (2009-2011)
- Honor student awards in every semester in undergraduate level, KFUPM, 2004-2009

Journal Publications

1. Abdulaziz Y. Barnawi and Ismail M. Keshta, “Energy Management in Wireless Sensor Networks Based on Naive Bayes, MLP, and SVM Classifications: A Comparative Study,” Journal of Sensors, vol. 2016, Article ID 6250319, 12 pages, 2016. **(ISI Journal**)
2. Farouq M. Aliyu, Yahya Osais, Ismail Keshta and Adel Binajjaj, “Maximizing Throughput of SW ARQ with Network Coding through Forward Error Correction” International Journal of Advanced Computer Science and Applications(IJACSA), 6(6), 2015. **(ISI Journal)**
3. Ismail Keshta, Mahmood Niazi and Mohammad Alshayeb, “Implementation of Requirements Management Specific Practices (SP1.3 and SP1.4) for Small and Medium Sized Software Development Organizations” SUBMITTED to Software Quality Journal (SQJ), 2016
4. Ismail Keshta, Mahmood Niazi and Mohammad Alshayeb, “Towards Implementation of CMMI Level 2 for Small and Medium Sized Software Development Organizations” SUBMITTED to Journal of Systems and Software (JSS), 2016

Conference Publications

1. Helmy, Tarek, Ismail Keshta, and Abdallah Rashed. "Performance Evaluation of System Resources Utilization with Sandboxing Applications." In Information Science and Applications, pp. 475-482. Springer Berlin Heidelberg, 2015. (**Indexed by SCOPUS and Springerlink**)
2. Barnawi, Abdulaziz Y., and Ismail M. Keshta. "Energy Management of Wireless Sensor Networks based on Multi-Layer Perceptrons." In European Wireless 2014; 20th European Wireless Conference; Proceedings of, pp. 1-6. VDE, 2014. (**Indexed by IEEE Xplore**)

Languages

Excellent communication skills in Arabic and English

Computer Skills

- Database Systems: MySQL
- Operating Systems: MS Windows, UNIX (Linux).

- Programming: Java, C, C++
- Software's used: LaTeX, MATLAB, Logisim, LogicWorks 4.0

References

Available upon request